

AD-A046 348

ARGONNE NATIONAL LAB ILL  
AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-T--ETC(U)  
APR 77 D J BINGAMAN

F/G 13/2

UNCLASSIFIED

CEEDO-TR-76-34

NL

1 OF 3  
AD  
A046348



AU No. —

DDC FILE COPY

ADA046348



CEEDO



CEEDO-TR-76-34

12  
B.S.

**AIR QUALITY ASSESSMENT MODEL FOR  
AIR FORCE OPERATIONS—SHORT-TERM  
EMISSION/DISPERSION COMPUTER  
CODE DOCUMENTATION**

ARGONNE NATIONAL LABORATORY  
9700 SOUTH CASS AVENUE  
ARGONNE, ILLINOIS 60439

APRIL 1977

FINAL REPORT FOR PERIOD  
1 JULY 1975-1 JANUARY 1977

DDC  
REF ID: A642114  
NOV 14 1977  
B

Approved for public release; distribution unlimited

**CIVIL AND ENVIRONMENTAL  
ENGINEERING DEVELOPMENT OFFICE**

(AIR FORCE SYSTEMS COMMAND)

TYNDALL AIR FORCE BASE  
FLORIDA 32403

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

<b>19 REPORT DOCUMENTATION PAGE</b>		READ INSTRUCTIONS BEFORE COMPLETING FORM	
18. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
CEEDO TR-76-34			
4. TITLE (and Subtitle) AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-TERM EMISSION/DISPERSION COMPUTER CODE DOCUMENTATION.		5. TYPE OF REPORT & PERIOD COVERED Final Report 1 Jul 75 to 1 Jan 77	
7. AUTHOR(S) Dorothy J. Bingaman		6. PERFORMING ORG. REPORT NUMBER	
8. CONTRACT OR GRANT NUMBER(S) Project Order 76-0003		9. PERFORMING ORGANIZATION NAME AND ADDRESS Argonne National Laboratory 9700 South Cass Avenue Argonne IL 60439	
11. CONTROLLING OFFICE NAME AND ADDRESS Civil Environmental & Engineering Development Office Det 1 HQ ADTC Tyndall AFB FL 32403		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F 1900 5A03	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12. 206p.		12. REPORT DATE Apr 1977	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		13. NUMBER OF PAGES 209	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) 9/312 739 7711 X 3026		15. SECURITY CLASS. (of this report) UNCLASSIFIED	
18. SUPPLEMENTARY NOTES Available in DDC.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE B NOV 14 1977	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Aircraft Assessment Airport Models Air Pollution Dispersion Model Computer Code		14 872-2264 4013 May Davis 8/970 2996	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The Air Force contracted with Argonne National Laboratory to develop a series of computer programs called the Air Quality Assessment Model (AQAM). The source emissions inventory routine of AQAM was designed to handle complex emission sources with emphasis on aircraft. A short term emission/dispersion model for hourly air quality predictions and a long term emission/dispersion model for monthly and annual predictions are also in AQAM.			
This report documents only the short term model. Flow charts, computer			

~~UNCLASSIFIED~~

~~SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)~~

Block 20 continued

listings, and brief descriptions of each subroutine are included. They are intended for readers with a computer background who wish to examine or alter the computer code.

ACCESSION for		
NTIS	NTIS Section <input checked="" type="checkbox"/>	
DDC	DDC Section <input type="checkbox"/>	
UNARMED/UNARMED		
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dis.	<input checked="" type="checkbox"/> OF SPECIAL	
A		

~~UNCLASSIFIED~~

~~SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)~~

PREFACE

This report documents work performed during the period 1 July 1975 through December 1976 by Argonne National Laboratory. The technical work for this effort was performed under the auspices of the Air Force Civil Engineering Center (AFSC) which on 8 April 1977, reorganized into Detachment 1 (CEEDO) HQ ADTC, Tyndall Air Force Base, Florida, 32403. Captain Dennis F. Naugle, CEEDO/ECA, managed the program.

This report has been reviewed by the Information Officer and is releasable to the National Technical Information Service (NTIS). At NTIS it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

*Dennis F. Naugle*

DENNIS F. NAUGLE, Capt, USAF, BSC  
Chief, Environmental Modeling  
Branch

*Signature of Dennis F. Naugle*

JOSEPH S. PIZZUTO, Col, USAF, BSC  
Commander

*Peter S. Daley*

PETER S. DALEY, Maj, USAF, BSC  
Chief, Environmental Assessment  
Research Division

*Peter A. Crowley*

PETER A. CROWLEY, Maj, USAF, BSC  
Director of Environics

## TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	1
Subroutine ABABAR	6
Subroutine ABLNAR	14
Subroutine ABPTAR	19
Subroutine ACSRCE	24
Subroutine AINE	39
BLOCK DATA	43
Subroutine CAVL	45
Subroutine CLASSE	65
Subroutine DEPART	68
Subroutine DIFERF	72
Subroutine EMISAR	75
Subroutine ENARAY	78
Subroutine INDINP/DEPINP	83
Program MAIN	90
Subroutine MAINS	98
Subroutine METHA	106
Subroutine METHB	111
Subroutine METHC	114
Subroutine METHD	117
Subroutine METHE	120
Subroutine OUTPUT	123
Subroutine PLRISE	128
Subroutine POLSOR	135
Subroutine PSEUDO	146
Subroutine QMOD	149
Subroutine READ	153
Subroutine RISE	158
Subroutine RRDIST	162
Subroutine SIGY/SIGCY	167
Subroutine SIGZ/SIGCZ	170
Subroutine SOURCE	175

TABLE OF CONTENTS (CONCLUDED)

	<u>Page</u>
<b>Subroutine STPOL1</b>	178
<b>Subroutine STPOL2</b>	185
<b>Subroutine TRAN</b>	190
<b>REFERENCES</b>	203

## SECTION I INTRODUCTION

Argonne National Laboratory (ANL) has developed an Air Quality Assessment Model (AQAM) for airbase operations under contract to the U.S. Air Force Civil Engineering Center (AFCEC) designed to simulate the emission of pollutants from sources on an airbase and the dispersion of these emissions in the atmosphere so as to enable calculation of pollutant concentrations over a grid of ground level receptors. These models are comprised of four physically separate computer codes, of which three must be operated by the user. The fourth code prepares a magnetic tape containing long term stability-time-wind roses for use by the long term climatological type air pollution model. This code is operated on request by the USAF Environmental Technical Applications Center in Washington, D.C. and the resultant magnetic tapes containing the climatological information is shipped to the user. The other three codes, developed by ANL, consist of the

- Source Inventory Model (SRCINV)
- Short Term Emission/Dispersion Model
- Long Term Emission/Dispersion Model

This report constitutes the computer code documentation for the second of these - the Short Term Emission/Dispersion Model. A separate computer code documentation manual (Reference 1) is available for SRCINV. Documentation for the Long Term Emission/Dispersion Model is currently being prepared and will be available shortly. A companion document to these reports - Operator's Guide (Reference 2) of the Air Quality Assessment Model for airbase operations - consists of a detailed discussion of the various functional parts of the computer programs and the input/output requirements. A second companion report (Reference 3) discusses the technical and theoretical basis underlying AQAM and presents and describes equations and algorithms used in the various AQAM sub-models.

The intended purpose of the present document is to provide a computer programmer with sufficient information so that he can study the code and make changes or modifications to it where required.

Table 1 contains a list of all routines contained in the Short Term Model in alphabetical order together with a brief description. More detailed descriptions of each routine, together with flow charts and computer code listings with comments that are intended to link listings to flow charts, are given on subsequent pages. It is hoped that this information, when combined with that given in References 1, 2, and 3, will enable a programmer to understand and make changes to the code when desired.

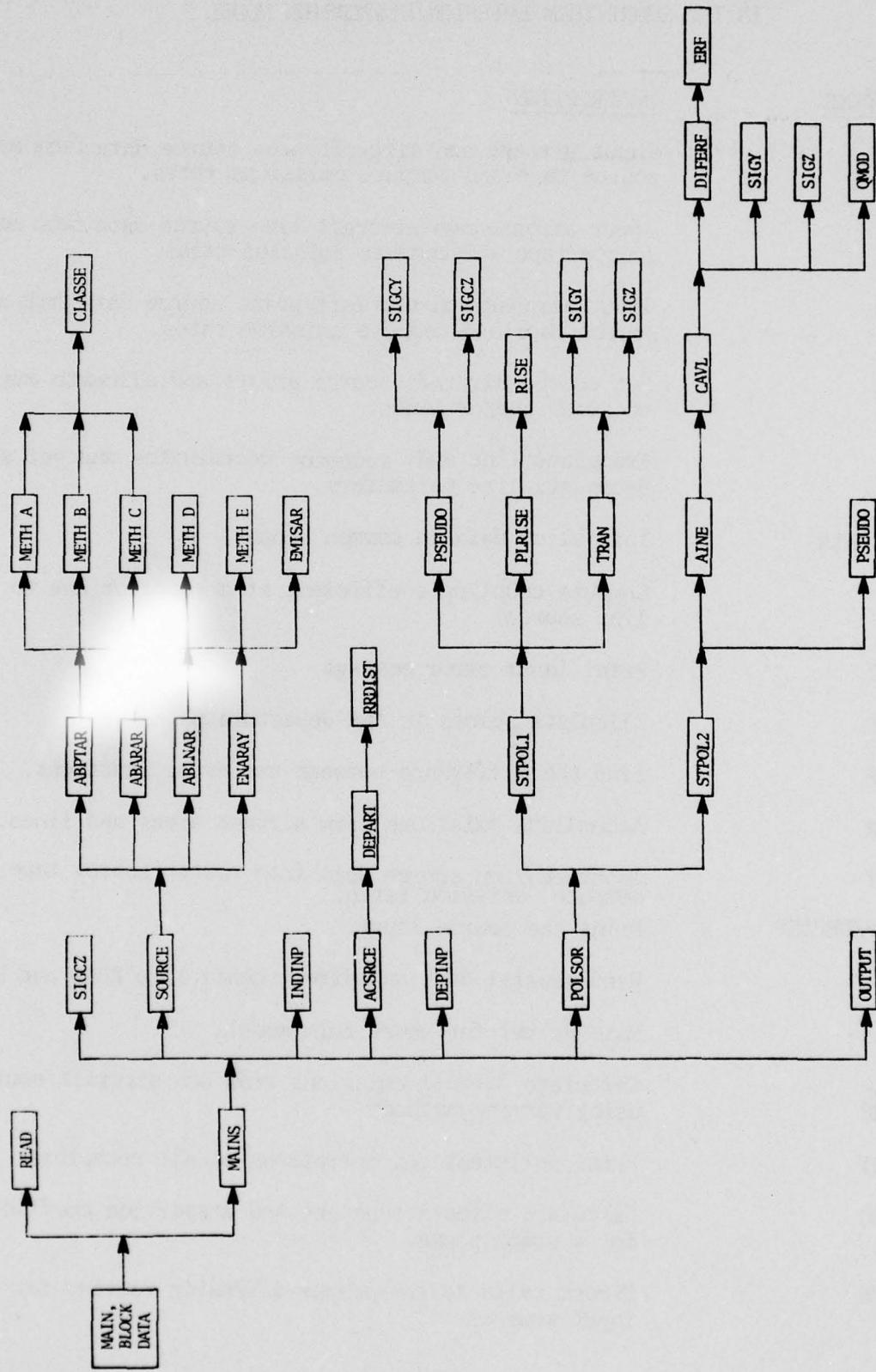


Figure 1. Schematic Flow Diagram of Short Term Model

TABLE 1. LIST OF ALL PROGRAMS AND SUBPROGRAMS  
IN THE SHORT TERM EMISSION/DISPERSION MODEL

<u>SUBROUTINE</u>	<u>DESCRIPTION</u>
ABARAR	Input airbase non-aircraft area source data from master source tape and compute emissions rates.
ABLNAR	Input airbase non-aircraft line source data from master source tape and compute emission rates.
ABPTAR	Input airbase non-aircraft point source data from master source tape and compute emission rates.
ACSRCE	Set up the aircraft source arrays and allocate emissions to areas and/or lines.
AINE	Translate line and receptor coordinates and set all necessary line parameters.
BLOCK DATA	Initialize data in common blocks.
CAVL	Compute coupling coefficient at a receptor due to a line source.
CLASSE	Print input error message.
DEPART	Calculate points in the departure path.
DIFERF	Find the difference between two error functions.
EMISAR	Accumulate emissions from airbase areas and lines.
ENARAY	Input environ source data from master source tape and compute emission rates.
INDINP/DEPINP	Print the source input.
MAIN	Read general data and direct control to READ and MAINS.
MAINS	Main driver for short term model.
METHA- METHE	Calculate diurnal emissions from non-aircraft sources using varying methods.
OUTPUT	Print pollutant concentrations at all receptors.
PLRISE	Calculate effective height and dispersion coefficients for a stack plume.
POLSOR	Direct calls to the proper diffusion routine for all input sources.

TABLE 1. LIST OF ALL PROGRAMS AND SUBPROGRAMS IN  
THE SHORT TERM EMISSION/DISPERSION MODEL (CONCLUDED)

PSEUDO	Call functions to find virtual distance from source to pseudo upwind point.
QMOD	Compute linear distribution of pollution along a runway.
READ	Read master source tape.
RISE	Calculate plume rise.
RRDIST	Calculate length of runway necessary for takeoff.
SIGY/SIGCY	Calculate horizontal dispersion or corresponding virtual distance.
SIGZ/SIGCZ	Calculate vertical dispersion or corresponding virtual distance.
SOURCE	Driver for non-aircraft emission routines.
STPOL1	Determine pollutant concentrations from point and area sources.
STPOL2	Determine pollutant concentrations from line sources.
TRAN	Calculate the coupling coefficient at a receptor due to a point or area source.

## SUBROUTINE ABARAR

### Purpose:

1. To read from the master source type all data needed to define airbase non-aircraft area sources.
2. To compute the emission rates due to evaporative hydrocarbons, space heating, off-road vehicles, and military and civilian vehicles.

### Input:

If the diurnal distribution cards are input, an additional parameter, IOPT, is read here to choose the method of distribution of those evaporative hydrocarbons not using the default of a uniform distribution.

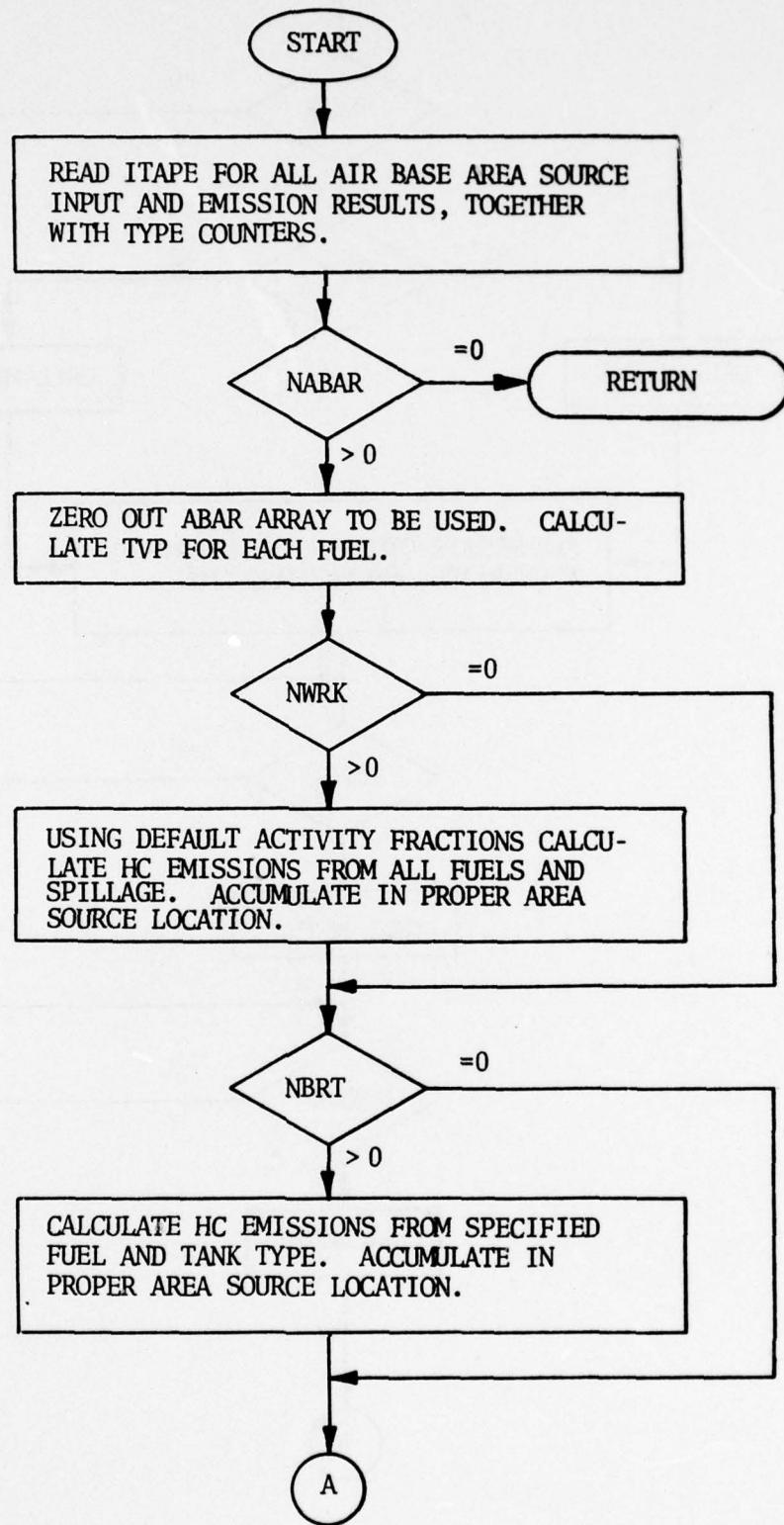
### Output:

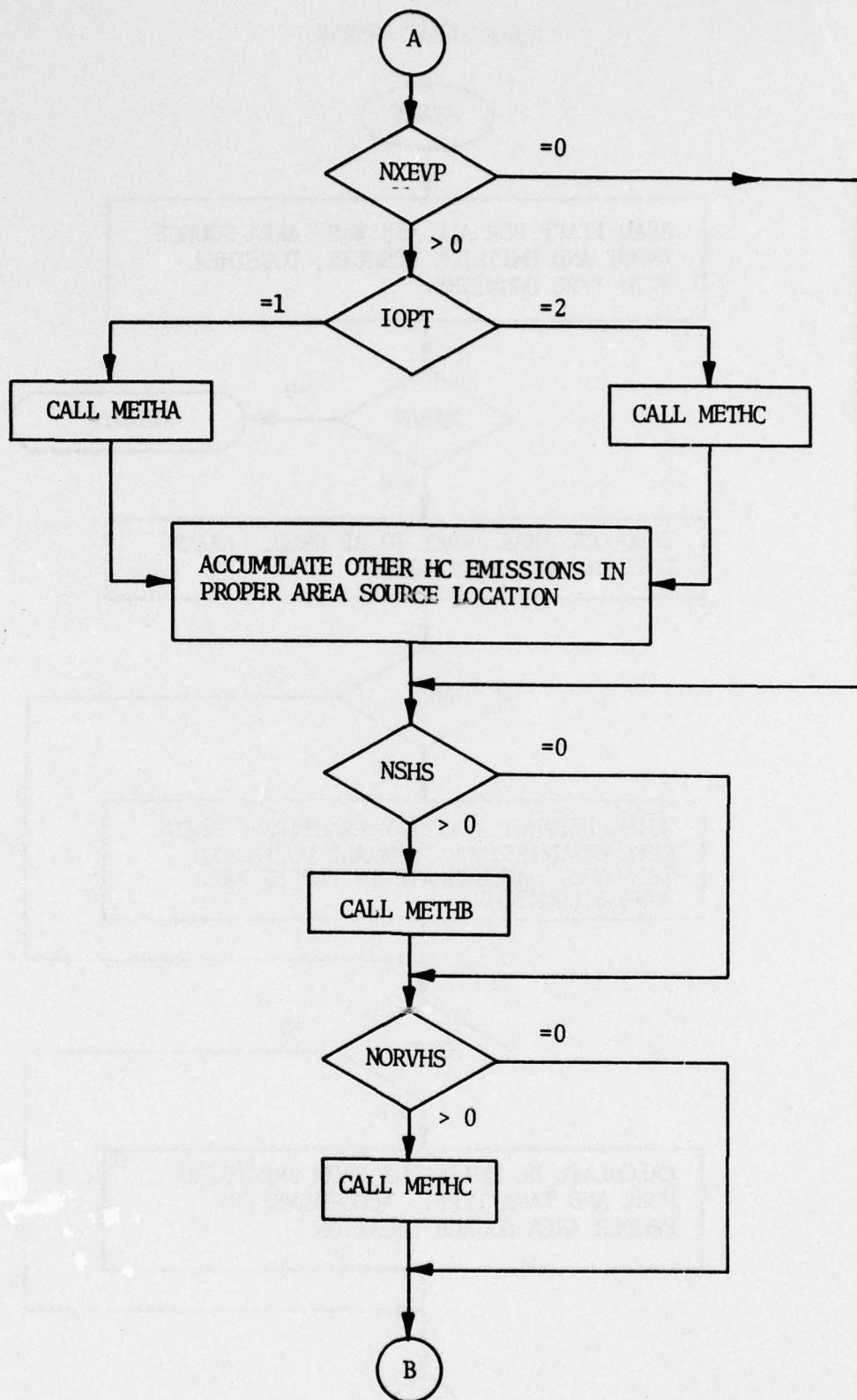
The array, ABAR, is filled with geometry and emission data for airbase non-aircraft area sources.

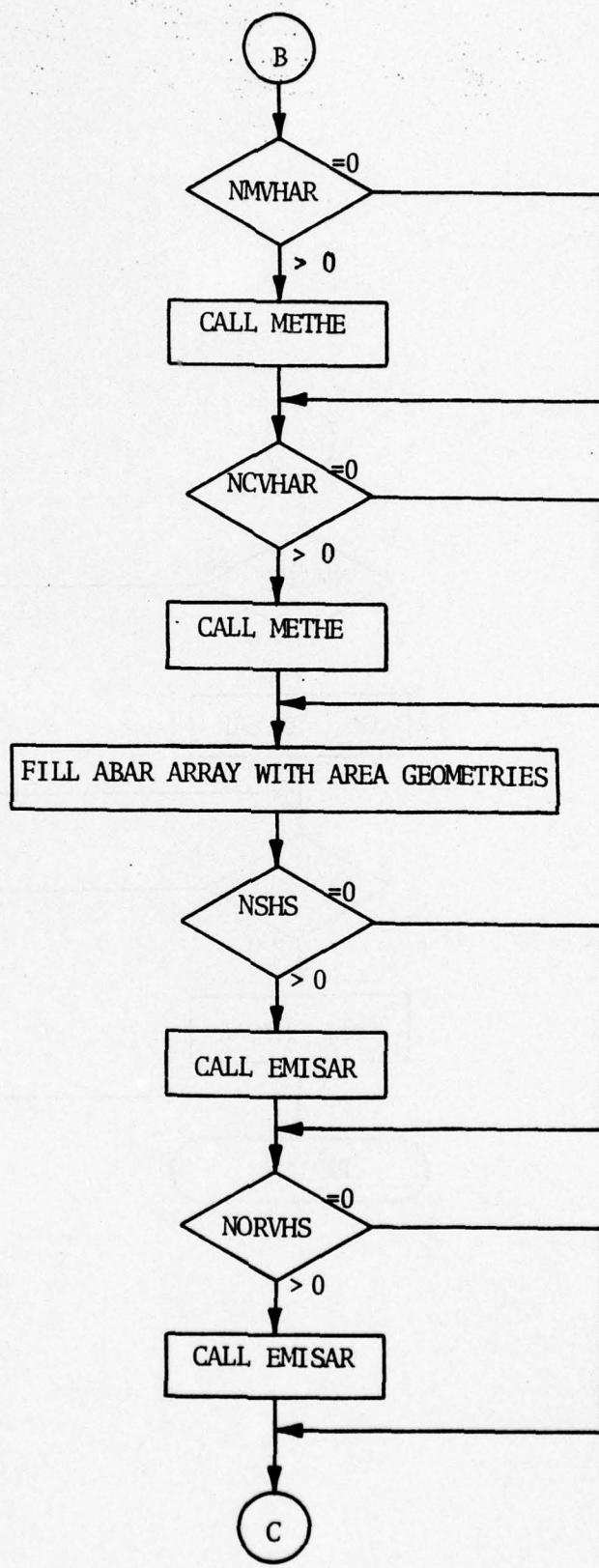
### Subroutines Called:

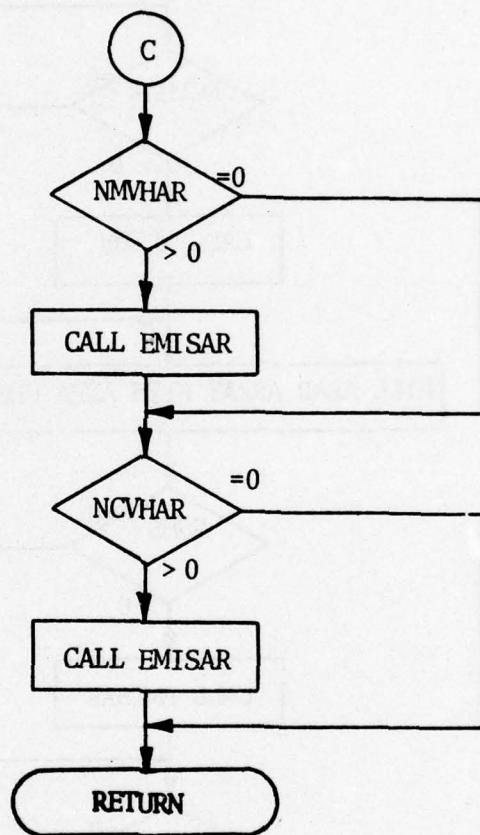
METHA, METHB, METHC, METHE, EMISAR

SUBROUTINE ABARAR









SUBFCUIINE ABARAR ABARR000  
 C ABARR001  
 C TIHS ROUTINE COMPUTES THE EMISSION RATES FOR ABARR002  
 C ALL AIRBASE AREAS ABARR003  
 C NWRK = NO. OF HYDROCARBON WORKING LOSSES ABARR004  
 C NEFT = NO. OF HYDROCARBON BREATHING LOSSES ABARR005  
 C NXEVP = NO. OF OTHER EVAPORATIVE HYDROCARBON SOURCES ABARR006  
 C NSHS = NO. OF SPACE HEATING SOURCES ABARR007  
 C NCRVHS = NO. OF OFF-ROAD VEHICLE SOURCES ABARR008  
 C NMVHAR = NO. OF MILITARY VEHICLE AREA SOURCES ABARR009  
 C NCVHAR = NO. OF CIVILIAN VEHICLE AREA SOURCES ABARR010  
 C ABARR011  
 C CCMMCN /PERIOD/ IMONTH,NODAYS, IDAY,IHR1,IHR2,IFLAG,JFLAG ABARR012  
 C CCMMCN / DEFALT / ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7) ABARR013  
 C COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13), ABARR014  
 . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13), ABARR015  
 . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1 ABARR016  
 C CCMMCN/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200) ABARR017  
 . ,LCC1,LOC2,NGECM,IET ABARR018  
 C COMMON/MONMET/TMBAR,WSMBAR,AMDMBR,DTMBAR ABARR019  
 C CCMMCN /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN, ABARR020  
 . NACPI,NACAR,NACLN,ENFT(16,100),ENAR(11,100),ENLN(14,20), ABARR021  
 . ABFT(16,150),ABAR(11,100),ABLN(14,100) ABARR022  
 C DIMENSION ABARGM(7,100),HCWRK(10,50),HCBRT(5,100),HCEVP(3,50), ABARR023  
 . FLHCUR(7),TVP(7) ABARR024  
 C EQUIVALENCE (SORGM(1),ABARGM(1)),(SORGM( 701),HCWRK(1)), ABARR025  
 . (SORGM(1201),HCBRT(1)),(SORGM(1701),HCEVP(1)) ABARR026  
 C LCC1=2 ABARR027  
 C LCC2=2 ABARR028  
 C NGECM=0 ABARR029  
 C IPT=0 ABARR030  
 C NSRCE=0 ABARR031  
 C I1=17 ABARR032  
 C I2=300 ABARR033  
 C ABARR034  
 C READ(ITAPE) NABAR,NTOT,NWRK,NBRT,NXEVP,NSHS,NORVHS, ABARR035  
 . NMVHAR,NCVHAR,NABARS,((ABARGM(I,N),I=1,7),N=1,NABAR), ABARR036  
 . ((HCWRK(I,N),I=1,10),N=1,NWRK), ABARR037  
 . ((HCEBT(I,N),I=1,5),N=1,NBFT), ABARR038  
 . ((HCEVP(I,N),I=1,3),N=1,NXEVP), ABARR039  
 . ((SOPCE(I,N),I=1,NTCT),N=1,NABARS) ABARR040  
 C IF (NABAR.EQ.0) GO TO 1100 ABARR041  
 C ABARR042  
 C ABARR043  
 C NHI=IHR2 ABARR044  
 C IF(IHR1.GT.IHR2) NHI=24+IHR2 ABARR045  
 C HFS=NHI-IHR1+1 ABARR046  
 C DC 10 N=1,NABAR ABARR047  
 C DC 10 I=1,NELTS ABARR048  
 C AEAR(I+5,N)=0.0 ABARR049  
 10 CCNTINUE ABARR050  
 C T=5./9.\*(TMBAR-32.0)+273. ABARR051  
 C DC 20 J=1,7 ABARR052  
 C TVP(J)=EXP(ALPHA(J)-BETA(J)/T) ABARR053  
 20 CCNTINUE ABARR054  
 C IF (NWRK.EQ.0) GO TO 100 ABARR055  
 C USING DEFAULT ACTIVITY FRACTIONS CALCULATE HC ABARR056  
 C EMISSIONS FROM ALL FUELS AND SPILLAGE. ABARR057  
 C ACCUMULATE IN ABAR ARFAY ABARR058  
 C ABARR059  
 C ABARR060  
 C DC 50 N=1,NWRK ABARR061

```

HC=0.
FRC=0.
DC 40 J=1,7
FLHOUR(J)=0.
DC 30 I=IHR1,NHI
II=I
IF(I.GT.24) II=I-24
30 FLHOUR(J)=FLHOUR(J)+FLHR(II,J)
FLHOUR(J)=FLHOUR(J)/HRS
FRC=FRC+FLHOUR(J)*FLMC(IMONTH,J)*FLDY(IDAY,J)
HC=HC+HCWRK(J+2,N)*TVP(J)*FLMO(IMONTH,J)*FLDY(IDAY,J)
. *FLHOUR(J)*7./DAYS
40 CONTINUE
FRC=FRC/4.*7./DAYS
C
J=HCWRK(2,N)
ABAR(7,J)=ABAR(7,J) + (HC + HCWRK(10,N) * FRC) *(1.E+6/3.6)
50 CCNTINUE
C
100 IF (NERT.EQ.0) GO TO 200
C CALCULATE HC EMISSIONS FROM SPECIFIED FUEL AND
C TANK TYPES. ACCUMULATE IN ABAR ARRAY
C
DC 110 N=1,NBRT
J=HCBRT(3,N)
EX=0.68
IF (HCBRT(4,N).EQ.2.) EX=0.70
HC=HCBRT(5,N)*(TVP(J)/(14.7-TVP(J)))**EX*(1.E+6/(3.6*24.*365.))
C
J=HCBRT(2,N)
ABAR(7,J)=ABAR(7,J)+HC
110 CCNTINUE
C
200 IF (NXEVP.EQ.0) GO TO 300
ICLASS=11C
NIEME=NELTS
NELTS=1
LCC1=3
NSRCE=NABARS
DC 210 N=1,NXEVP
DC 210 I=1,3
SCRCE(I,NABARS+N)=HCEVE(I,N)
210 CONTINUE
ICPT=1
IF (JFLAG.EQ.0) READ 2,IOPT
2 FORMAT(I4)
GC TO (220,230),IOPT
C
220 CALL METHA(NXEVP,SORCE,I1,I2,ICLASS)
GC TO 240
230 CALL METHC(NXEVP,SORCE,I1,I2,ICLASS)
C
240 DC 250 N=1,NXEVP
C
C ACCUMULATE OTHER EVAPORATIVE HC EMISSIONS IN ABAR ARRAY
C
J=HCEVE(2,N)
ABAR(7,J)=ABAR(7,J)+SCRCE(3,NABARS+N)
250 CONTINUE
C
NELTS=NTEMP
NSRCE=0

```

```

      LCC1=2                                ABARR124
C      300 IF (NSHS.EQ.0) GO TO 400          ABARR125
      ICLASS=111                            ABARR126
      CALL METHB(NSHS,SORCE,I1,I2,ICLASS)  ABARR127
C      400 IF (NCFVHS.EQ.0) GO TO 500        ABARR128
      ICLASS=112                            ABARR129
      CALL METHC(NCFVHS,SOPCE,I1,I2,ICLASS) ABARR130
C      500 IF (NMVHAF.EQ.0) GO TO 600        ABARR131
      CALL METHE(NMVHAF,SORCE,MLMO,VHMLDY,VHMLHR,I1,I2) ABARR132
C      600 IF (NCVHAR.EQ.0) GO TO 700        ABARR133
      CALL METHE(NCVHAR,SORCE,CVABMO,CVABDY,CVABHR,I1,I2) ABARR134
C      ****EMISSIONS ARE NOW IN MICROGRAMS/SEC ABARR135
C      FILL ABAR ARRAY WITH AREA GEOMETRIES ABARR136
C
      700 DO 710 N=1,NABAR                  ABARR137
      DC 710 I=1,5                          ABARR138
      ABAR(I,N)=ABAEGM(I+2,N)              ABARR139
      710 CCNTINUE                         ABARR140
C      FILL ABAR ARRAY WITH THE NON-EVAP HC EMISSION DATA ABARR141
C
      I1=11                                ABARR142
      I2=100                               ABARR143
      NSRCF=0                             ABARR144
      LCC1=5                             ABARR145
      IF (NSHS.FQ.0) GO TO 800            ABARR146
      CALL EMISAR(NSHS,ABAR,I1,I2)        ABARR147
C      800 IF (NORVHS.EQ.0) GO TO 900        ABARR148
      CALL EMISAR(NCFVHS,ABAR,I1,I2)        ABARR149
C      900 IF (NMVHAF.EQ.0) GO TO 1000       ABARR150
      CALL EMISAR(NMVHAR,ABAR,I1,I2)        ABARR151
C      1000 IF (NCVHAR.EQ.0) GO TO 1100       ABARR152
      CALL EMISAR(NCVHAR,ABAR,I1,I2)        ABARR153
C      1100 CCNTINUE                      ABARR154
      RETURN                               ABARR155
      END                                  ABARR156
                                         ABARR157
                                         ABARR158
                                         ABARR159
                                         ABARR160
                                         ABARR161
                                         ABARR162
                                         ABARR163
                                         ABARR164
                                         ABARR165
                                         ABARR166
                                         ABARR167
                                         ABARR168

```

## SUBROUTINE ABLNAR

### Purpose:

1. To read from the master source tape all data needed to define airbase non-aircraft line sources.
2. To compute the emission rates due to military and civilian vehicle line and other airbase line activities.

### Input:

If the diurnal distribution cards are input, an additional parameter, IMETH, is input here to choose the method of distribution of emissions from those other airbase line activities not using the default of a uniform distribution.

### Output:

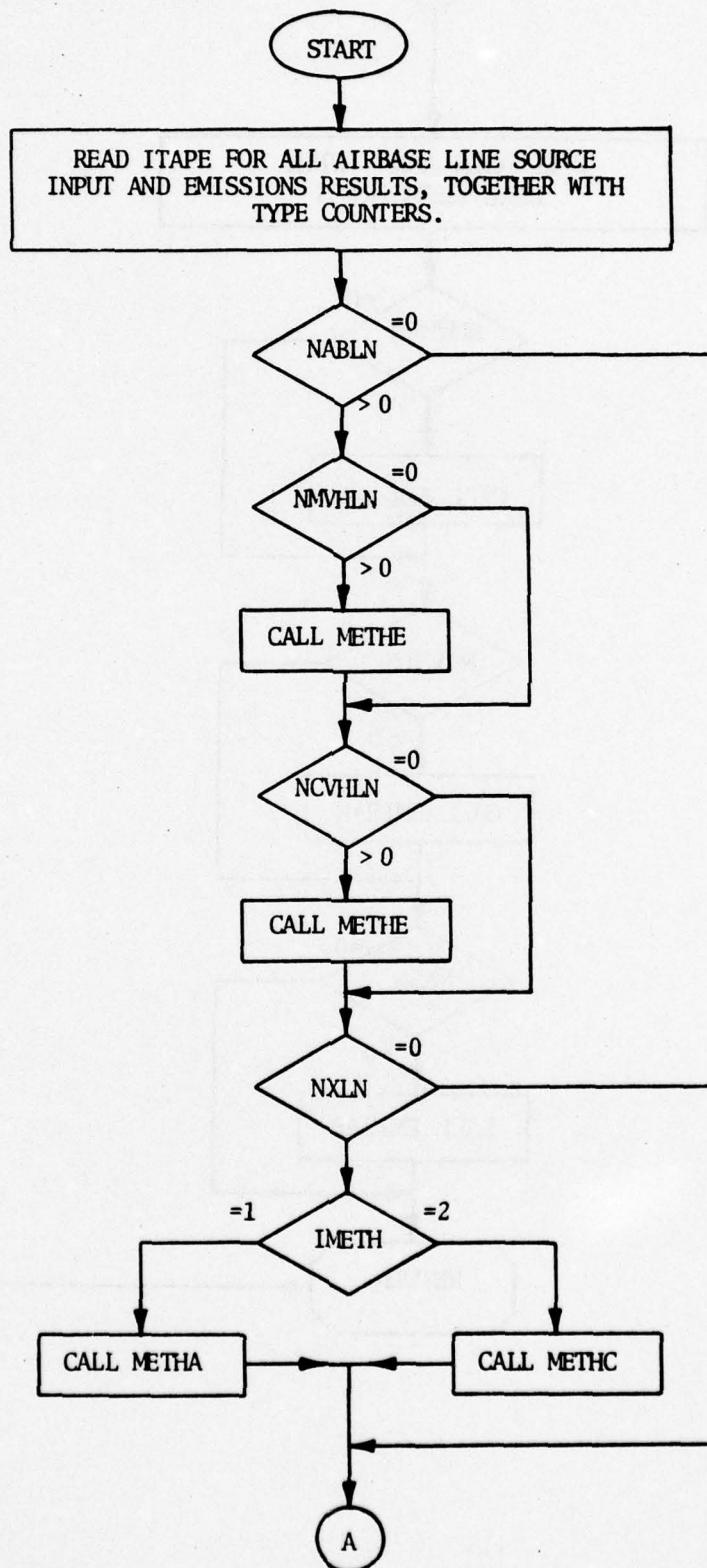
The array, ABLN, is filled with geometry and emission data for non-aircraft line sources.

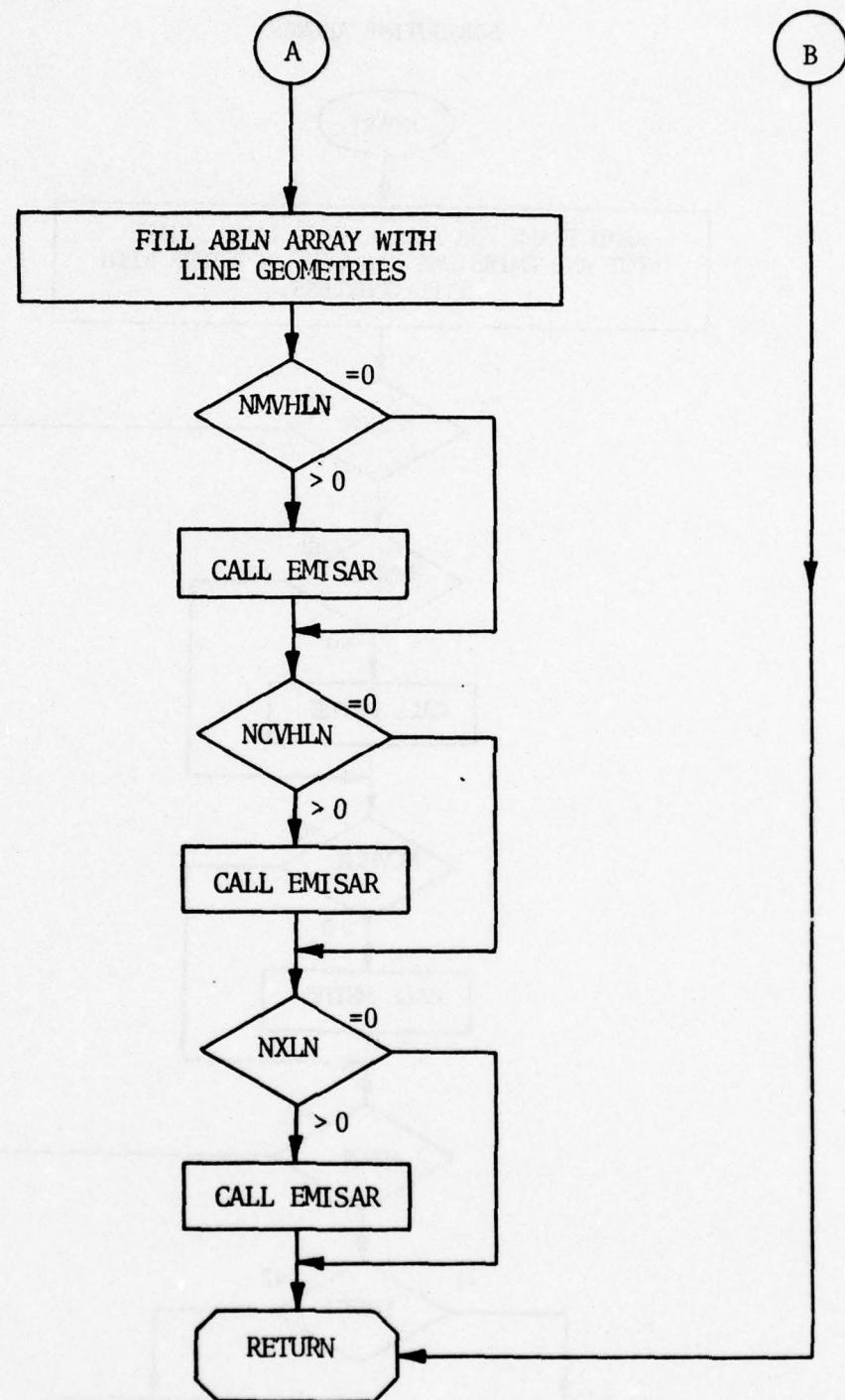
### Subroutines

#### Called:

METHA, METHC, METHE, EMISAR

SUBROUTINE ABLNR





```

SUBROUTINE ABLNAR
C
C THIS ROUTINE COMPUTES THE EMISSION RATES FOR ALL
C AIRBASE LINES
C      NMVHLN = NO. OF MILITARY LINE ACTIVITIES
C      NCVHLN = NO. OF CIVILIAN LINE ACTIVITIES
C      NXLN   = NO. OF OTHER AIR BASE LINE ACTIVITIES
C
COMMON / DEFAILT / ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7) ABLNR000
COMMON / PERIOD / IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG ABLNR001
COMMON / JUNK / DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200) ABLNR002
C      LOC1,LOC2,NGEOM,IPT ABLNR003
COMMON / DSTRET / ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13), ABLNR004
C      VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13), ABLNR005
C      CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1 ABLNR006
COMMON / SPCE / NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLNS, ABLNR007
C      NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20), ABLNR008
C      ABPT(16,150),ABAR(11,100),ABLN(14,100) ABLNR009
LOC1=2 ABLNR010
LOC2=2 ABLNR011
I1=17 ABLNR012
I2=300 ABLNR013
NGEOM=0 ABLNR014
IPT=0 ABLNR015
NSRCE=0 ABLNR016
FAD (ITAPE) NABLNS,NTOT,NMVHLN,NCVHLN,NXLN,NABLNS, ABLNR017
C      ((SORGM(I,N),I=1,10),N=1,NABLNS), ABLNR018
C      ((SOPCE(I,N),I=1,NTCT),N=1,NABLNS) ABLNR019
C
IF (NABLNS.EQ.0) GO TO 600 ABLNR020
IF (NMVHLN.EQ.0) GO TO 100 ABLNR021
C
CALL METHE(NMVHLN,SORCE,VHMLMO,VHMLDY,VHMLHR,I1,I2) ABLNR022
C
100 IF (NCVHLN.EQ.0) GO TO 200 ABLNR023
C
CALL METHE(NCVHLN,SORCE,CVABMO,CVABDY,CVABHR,I1,I2) ABLNR024
C
200 IF (NXLN.EQ.0) GO TO 300 ABLNR025
ICLASS=117 ABLNR026
C
IMETH=1 ABLNR027
IF (JFLAG.EQ.0) READ 1,IMETH ABLNR028
1 FFORMAT(I4) ABLNR029
GC TO (210,220),IMETH ABLNR030
C
210 CALL METHA(NXLN,SORCE,I1,I2,ICLASS) ABLNR031
GC TO 300 ABLNR032
C
220 CALL METHC(NXLN,SORCE,I1,I2,ICLASS) ABLNR033
C
***** EMISSIONS ARE NOW IN MICROGRAMS/SEC ABLNR034
C      FILL ABLN ARRAY WITH LINE GEOMETRIES ABLNR035
C
300 DC 320 N=1,NABLNS ABLNR036
DC 310 I=1,8 ABLNR037
ABLN(I,N)=SORGM (I+2,N) ABLNR038
310 CCNTINUE ABLNR039
DC 320 I=1,NPLTS ABLNR040
ABLN(I+8,N)=0.0 ABLNR041
320 CONTINUE ABLNR042
C

```

```
C      FILL ABLN ARRAY WITH LINE EMISSION DATA          ABLNR062
C
C      NSRCE=0          ABLNR063
LOC1=8          APLNR064
I1=14          ABLNR065
I2=100          ABLNR066
IF (NMVHLN.EQ.0) GO TO 400          ABLNR067
CALL EMISAR(NMVHLN,ABLN,I1,I2)          ABLNR068
C
400 IF (NCVHLN.EQ.0) GO TO 500          ABLNP070
CALL EMISAR(NCVHLN,ABLN,I1,I2)          ABLNP071
C
500 IF (NXLN.EQ.0) GO TO 600          ABLNP072
CALL EMISAR(NXLN,ABLN,I1,I2)          ABLNR073
C
600 CCONTINUE          ABLNP074
RETURN          ABLNP075
END          ABLNP076
          ABLNP077
          AFLNR078
          ABLNP079
```

## SUBROUTINE ABPTAR

### Purpose:

1. To read from the master source tape all data needed to define airbase non-aircraft point sources.
2. To compute the emission rates due to training fires, test cells, run-up stands, power plants, incinerators, storage tanks and other airbase point source activities.

### Input:

None

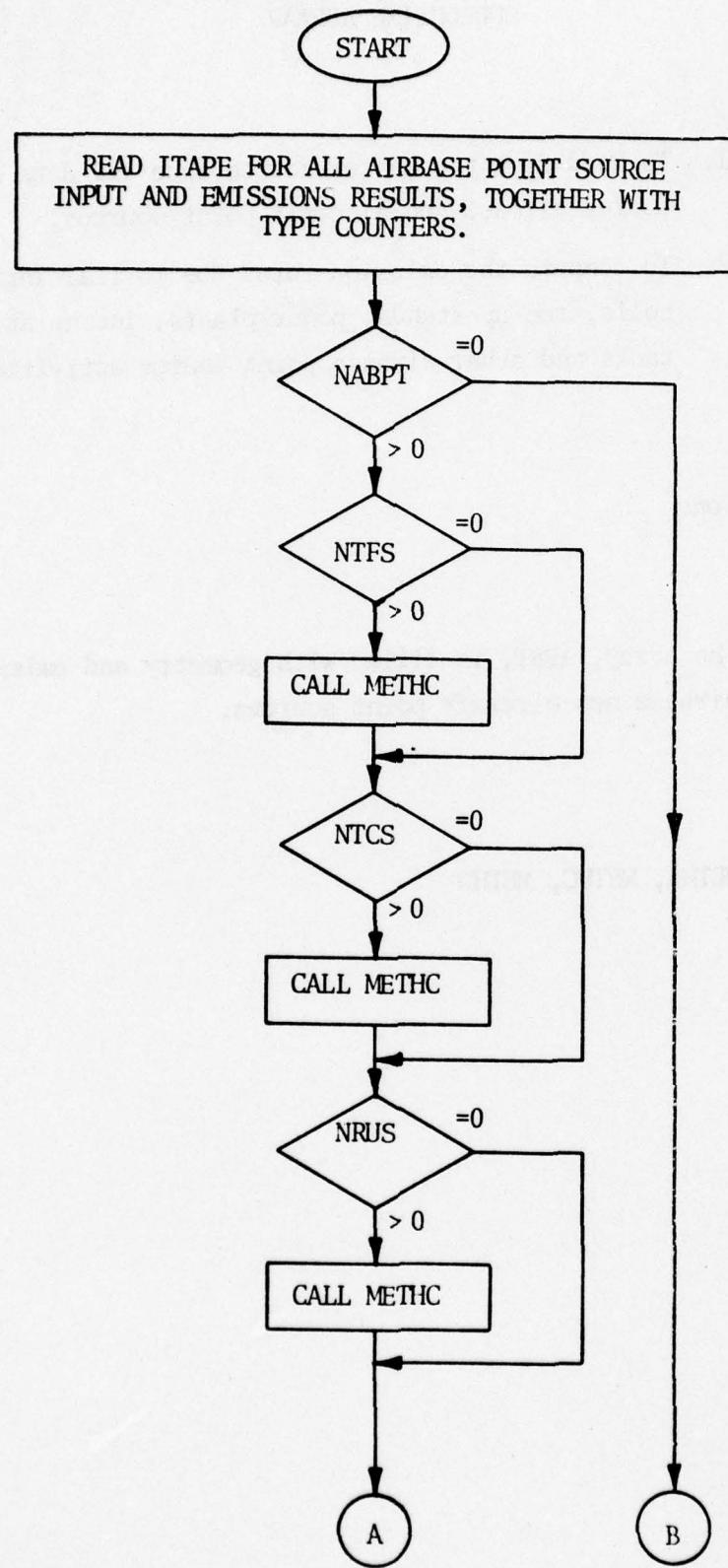
### Output:

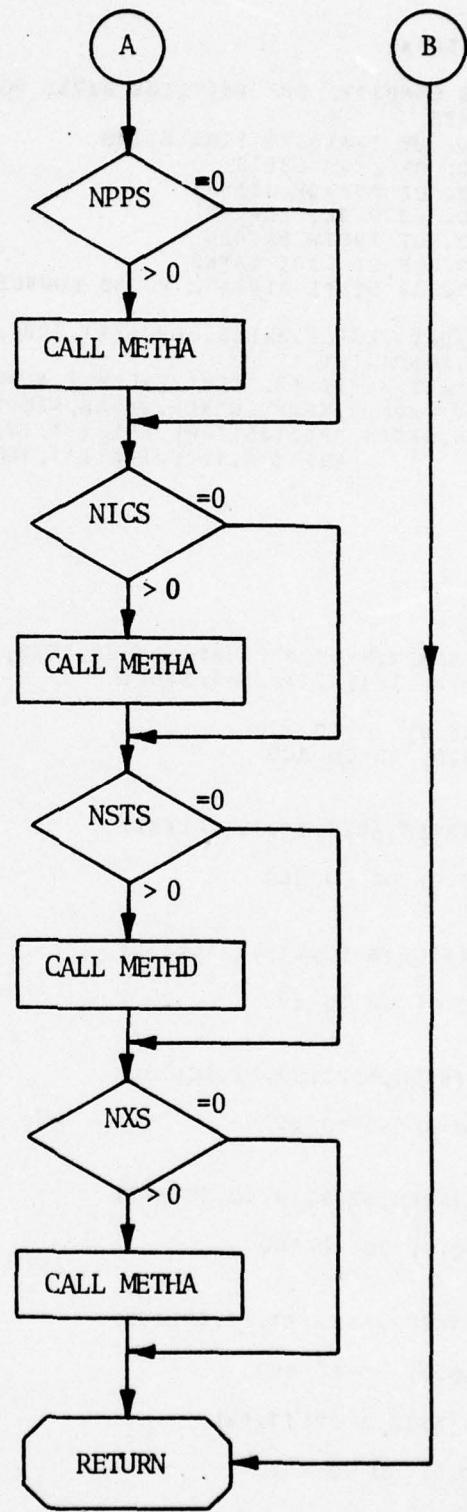
The array, APBT, is filled with geometry and emission data for airbase non-aircraft point sources.

### Subroutines Called:

METHA, METHC, METHD

SUBROUTINE ABPTAR





```

SUBROUTINE ABPTAR

C THIS ROUTINE COMPUTES THE EMISSION RATES FOR ALL
C AIRBASE POINTS
C      NTFS = NO. OF TRAINING FIRE SITES
C      NTCS = NO. OF TEST CELLS
C      NRUS = NO. OF RUN-UP STANDS
C      NPPS = NO. OF POWER PLANTS
C      NICS = NO. OF INCINERATORS
C      NSTS = NO. OF STORAGE TANKS
C      NXS = NO. OF OTHER AIRBASE POINT SOURCES
C
C      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
C      .,LOC1,LOC2,NGEOM,IPT
C      COMMON / DEFAUT / ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
C      COMMON / SRCE / NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABIN,
C      .,NACFT,NACAR,NACLN,ENFT(16,100),ENAR(11,100),ENLN(14,20),
C      .,ABFT(16,150),ABAR(11,100),ABLN(14,100)
C
C      LCC1=10
C      LCC2=11
C      NGEOM=9
C      IPT=1
C      NSRCE=0
C      I1=16
C      I2=200
C      READ(ITAPE) NABPT,NTOT,NTFS,NTCS,NRUS,NPPS,NICS,NSTS,NXS,
C      .,((SCFCE(I,N),I=1,NTOT),N=1,NABPT)
C
C      IF (NABFT.EQ.0) GO TO 700
C      IF (NTFS.EQ.0) GO TO 100
C      ICCLASS=101
C
C      CALL METHC(NTFS,ABPT,I1,I2,ICCLASS)
C
C      100 IF (NTCS.EQ.0) GO TO 200
C      ICCLASS=102
C
C      CALL METHC(NICS,ABPT,I1,I2,ICCLASS)
C
C      200 IF (NRUS.EQ.0) GO TO 300
C      ICCLASS=103
C
C      CALL METHC(NRUS,ABPT,I1,I2,ICCLASS)
C
C      300 IF (NPPS.EQ.0) GO TO 400
C      ICCLASS=104
C
C      CALL METHA(NPPS,ABPT,I1,I2,ICCLASS)
C
C      400 IF (NICS.EQ.0) GO TO 500
C      ICCLASS=105
C
C      CALL METHA(NICS,ABPT,I1,I2,ICCLASS)
C
C      500 IF (NSTS.EQ.0) GO TO 600
C
C      CALL METHD(NSTS,ABPT,I1,I2)
C
C      600 IF (NXS.EQ.0) GO TO 700
C      ICCLASS=107
C
C      CALL METHA(NXS,ABPT,I1,I2,ICCLASS)
C
C      ABPTR000
C      ABPTR001
C      ABPTR002
C      ABPTR003
C      ABPTR004
C      ABPTR005
C      ABPTR006
C      ABPTR007
C      ABPTR008
C      ABPTR009
C      ABPTR010
C      ABPTR011
C      ABPTR012
C      ABPTR013
C      ABPTR014
C      ABPTR015
C      ABPTR016
C      ABPTR017
C      ABPTR018
C      ABPTR019
C      ABPTR020
C      ABPTR021
C      ABPTR022
C      ABPTR023
C      ABPTR024
C      ABPTR025
C      ABPTR026
C      ABPTR027
C      ABPTR028
C      ABPTR029
C      ABPTR030
C      ABPTR031
C      ABPTR032
C      ABPTR033
C      ABPTR034
C      ABPTR035
C      ABPTR036
C      ABPTR037
C      ABPTR038
C      ABPTR039
C      ABPTR040
C      ABPTR041
C      ABPTR042
C      ABPTR043
C      ABPTR044
C      ABPTR045
C      ABPTR046
C      ABPTR047
C      ABPTR048
C      ABPTR049
C      ABPTR050
C      ABPTR051
C      ABPTR052
C      ABPTR053
C      ABPTR054
C      ABPTR055
C      ABPTR056
C      ABPTR057
C      ABPTR058
C      ABPTR059
C      ABPTR060
C      ABPTR061

```

C  
700 RETURN  
END

ABPTR062  
ABPTR063  
ABPTR064

## SUBROUTINE ACSRCE

### Purpose:

To set up the aircraft source arrays to be used by the dispersion routines for calculating ground level concentrations.

### Input:

Basic aircraft data, airbase activity data, points in arrival-departure paths and in training flight paths, meteorological conditions, time period of calculation.

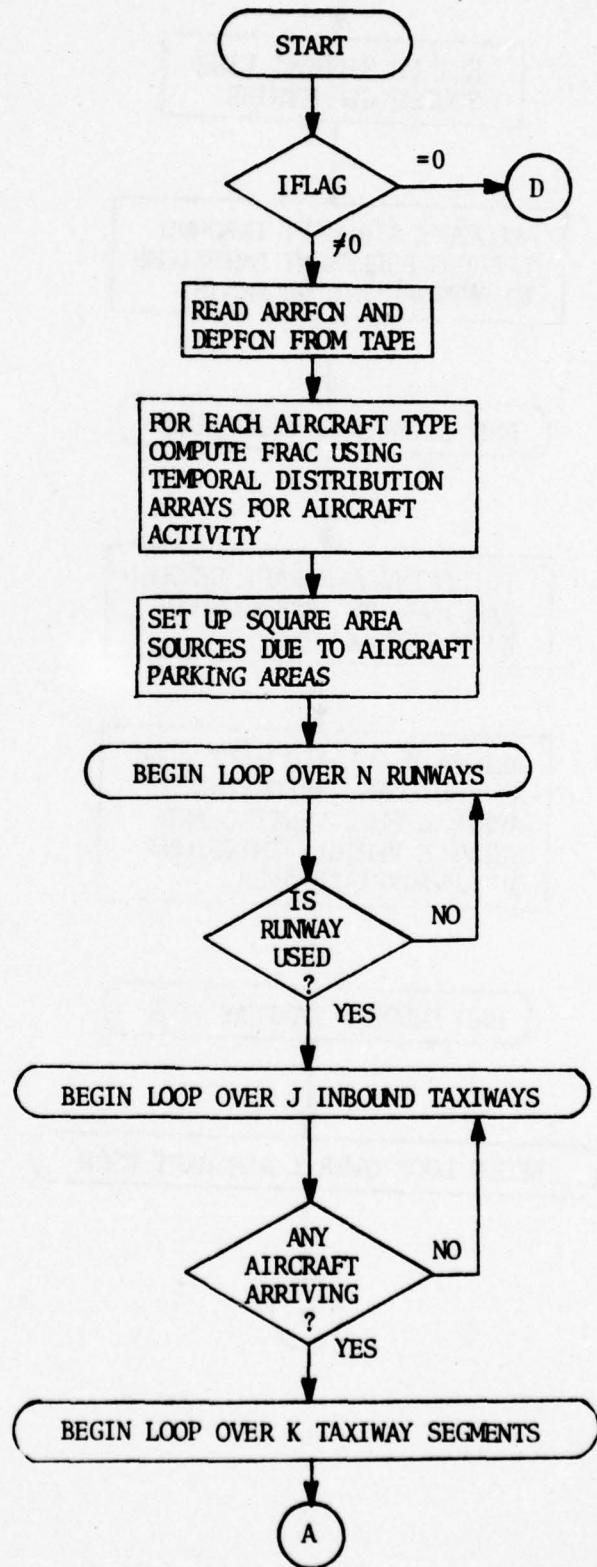
### Output:

The arrays ACPT, ACLN and ACAR to contain all source information necessary to calculate dispersion and pollutant concentrations.

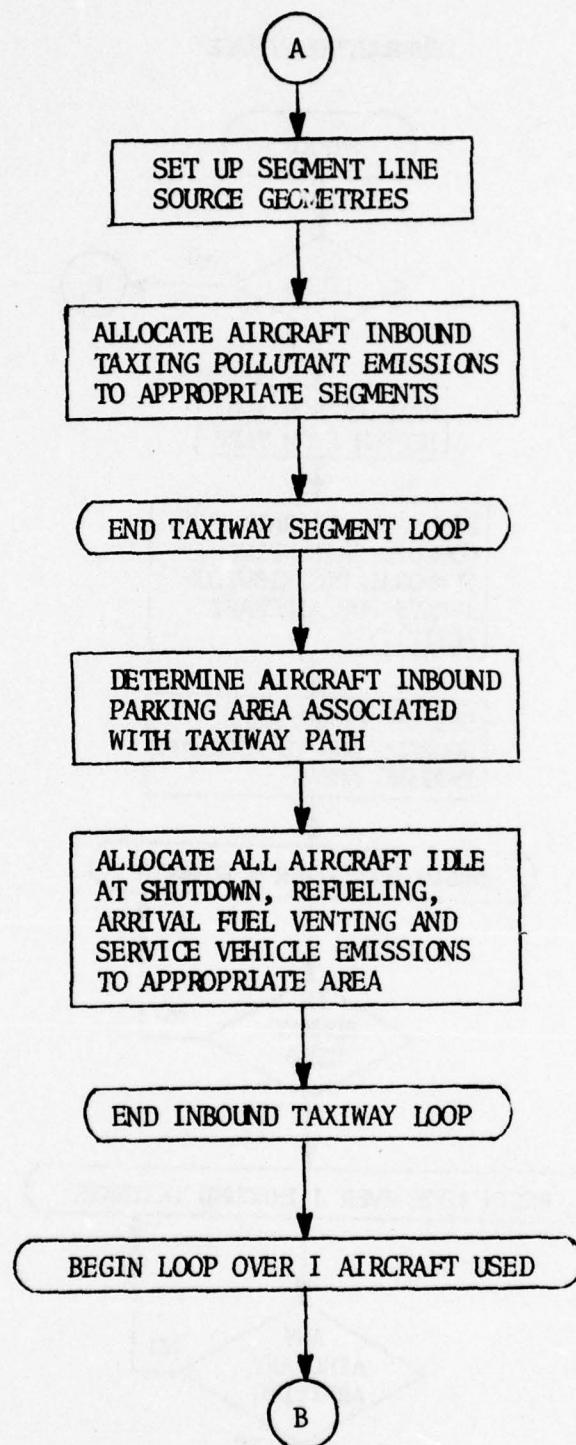
### Subroutine Called:

DEPART

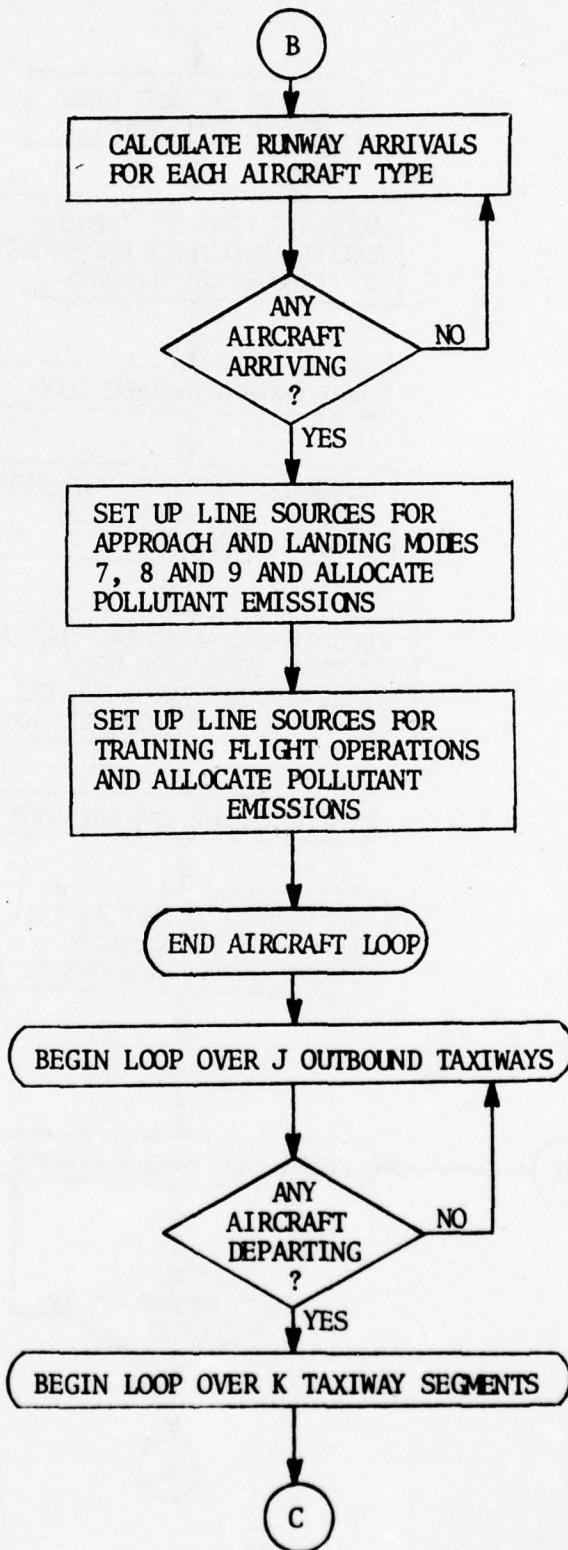
SUBROUTINE ACSRCE



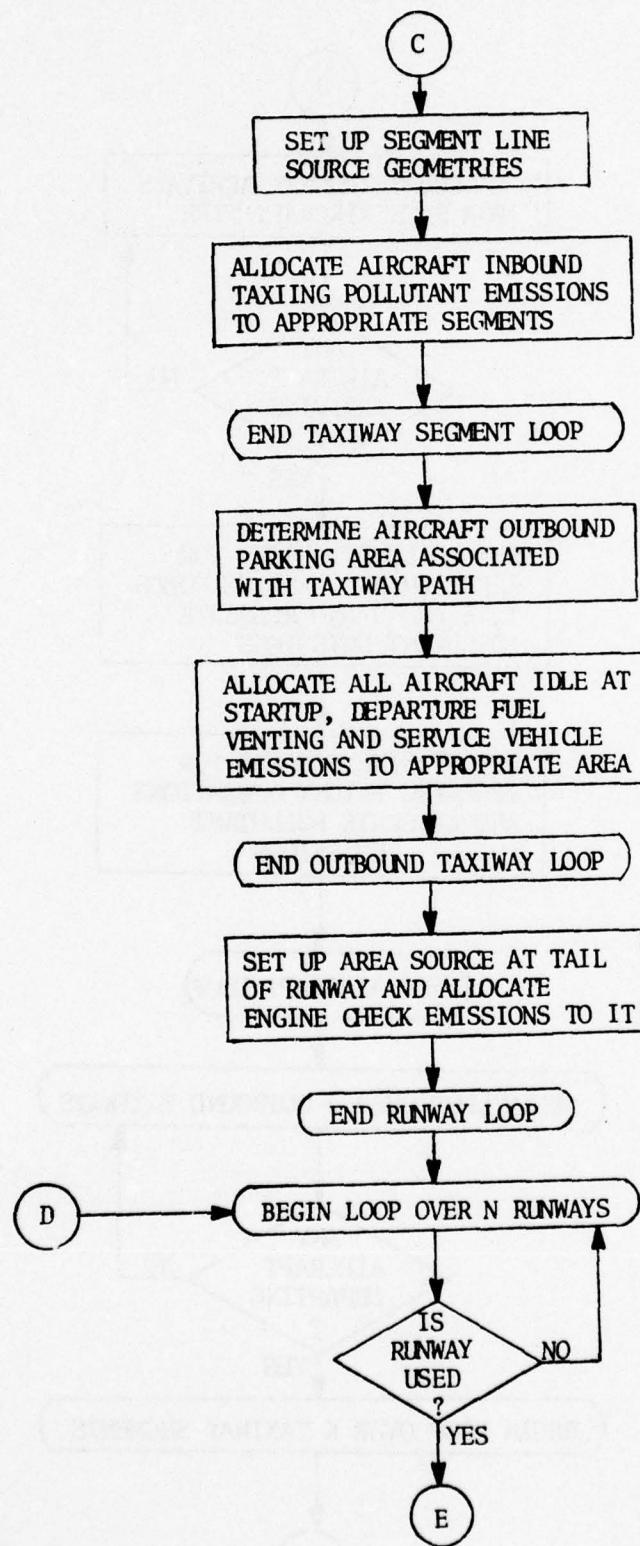
SUBROUTINE ACSRCE (Cont'd.)



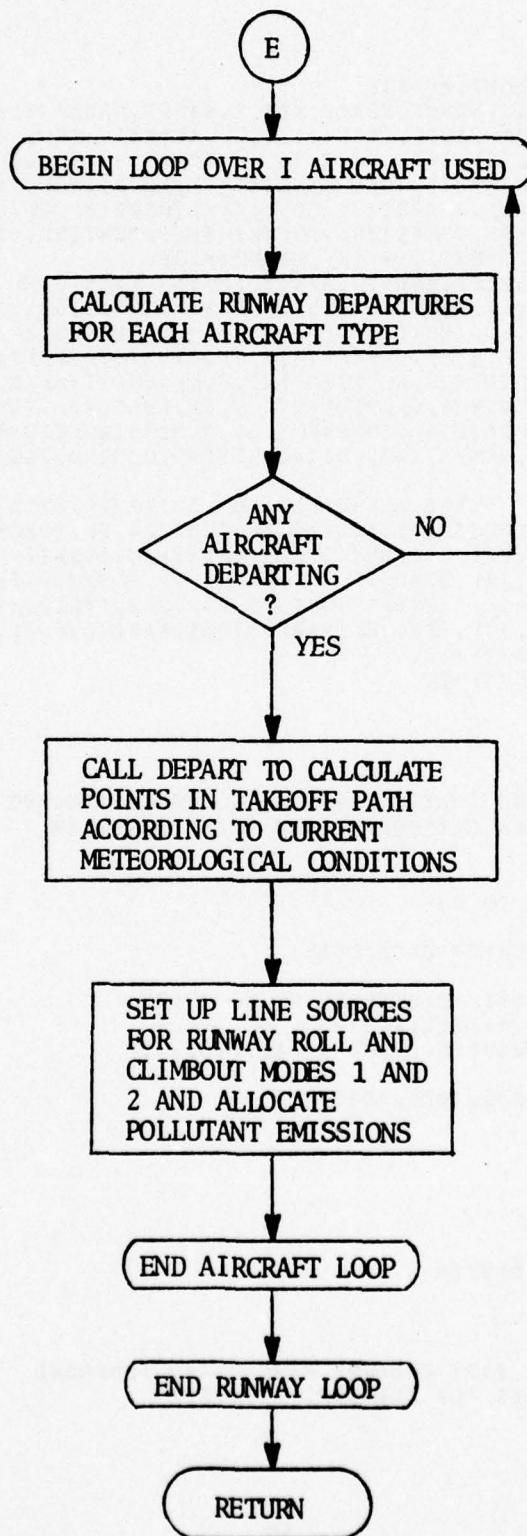
SUBROUTINE ACSRCE (Cont'd.)



SUBROUTINE ACSRCE (Cont'd.)



SUBROUTINE ACSRCE (Cont'd.)



```

SUBROUTINE ACSRCE                                ACSRC000
C
C THIS ROUTINE SETS UP THE AIRCRAFT SOURCE ARRAYS   ACSRC001
C AND ALLOCATES THE POLLUTANT EMISSIONS TO THE    ACSRC002
C APPROPRIATE AREA OR LINE                         ACSRC003
C
C
REAL LDSPD                                     ACSRC004
INTEGER ENGNO                                  ACSRC005
COMMON /RECPT/ MRECPT,MAXFIL                   ACSRC006
COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABL,NAACP,   ACSRC007
. NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),   ACSRC008
. ABAF(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)   ACSRC009
COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),   ACSRC010
. LDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),   ACSRC011
. COSPD2(8),SRTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),   ACSRC012
. APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2)   ACSRC013
COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),   ACSRC014
. ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),   ACSRC015
. DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),   ACSRC016
. ACSPIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),   ACSRC017
. IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),   ACSRC018
. IOBSEG(16,8,6),IOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),   ACSRC019
. PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8)   ACSRC020
COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,   ACSRC021
1 TEMK                                         ACSRC022
COMMON /DEFLT/ ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)   ACSRC023
COMMON /DSTRET/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),   ACSRC024
. VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),   ACSRC025
. CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1   ACSRC026
COMMON /PERIOD/ IMO,NODAYS,IDX,IHR1,IHR2,IFLAG,JFLAG   ACSRC027
DIMENSION IACAR(2,18),FRAC(8),PARFCT(18),APARSQ(6,3),NQ(25)   ACSRC028
XP(XO,YC,W)=YC*SIN(W)+XO   ACSRC029
YP(YO,YC,W)=YC*COS(W)+YO   ACSRC030
DAYS=NODAYS   ACSRC031
NT=NPLTS+5   ACSRC032
IWIND=29+IWD   ACSRC033
C
C AN IFLAG OF 0 MEANS THAT ALL AIRCRAFT SOURCES EXCEPT   ACSRC034
C FOR RUNWAY ROLL AND CLIMBOUT MODES 1 AND 2 REMAIN   ACSRC035
C UNCHANGED   ACSRC036
C
IF(IFLAG.EQ.0) GO TO 69   ACSRC037
C
C READ ARRFCN AND DEPFCN FROM TAPE   ACSRC038
C
IF (IWD.GE.1.AND.IWD.LE.MAXFIL) GO TO 1000   ACSRC039
PRINT 9000,MRECPT,MAXFIL,IWD   ACSRC040
9000 FORMAT(29HFILE REQUEST ERROR IN ACSRCE,3I5)   ACSRC041
GO TO 1040   ACSRC042
1000 IF (MRECPT-IWD) 1010,1030,1020   ACSRC043
1010 READ (30)   ACSRC044
MRECPT=MRECPT+1   ACSRC045
GO TO 1000   ACSRC046
1020 REWIND 30   ACSRC047
MRECPT=1   ACSRC048
GO TO 1000   ACSRC049
1030 READ (30) ARRFCN,DEPFCN   ACSRC050
MRECPT=MRECPT+1   ACSRC051
1040 CCNTINUE   ACSRC052
C
C FOR EACH AIRCRAFT TYPE COMPUTE FRAC USING TEMPORAL   ACSRC053
C DISTRIBUTION ARRAYS FOR AIRCRAFT ACTIVITY   ACSRC054

```

```

C
NHI=IHR2
IF(IHR1.GT.IHR2) NHI=24+IHR2
HRS=NHI-IHR1+1
DC 5 I=1,NACTYP
HRFRC=0.
DC 4 JJ=IHR1,NHI
J=JJ
IF(JJ.GT.24) J=JJ-24
4 HRFRC=HFFRC+ACHR(J,I)
HFFRC=HFFRC/HFS
FRAC(I)=ACMO(IMO,I)*ACDY(IDY,I)*HRFRC*7.0/DAYS*(1.E+6/3.6)
5 CONTINUE
8 NACPI=0
NB=0
NC=0
NZ=0
C
C SET UP SQUARE AREA SOURCES DUE TO AIRCRAFT PARKING AREAS
C
DO 1 L=1,NPKAR
NSQ=NPASQ(L)
SFARSQ=0.0
DO 2 J=1,NSQ
NB=NB+1
ACAF(1,NB)=PAREA(L,J,1)
ACAF(2,NB)=PAFEA(L,J,2)
ACAR(3,NB)=ACLNDZ/2.
ACAR(4,NB)=PAREA(L,J,3) *1000.
AEAFSC(L,J)=ACAR(4,NB) ** 2
SPARSQ = SPARSQ + APARSQ(L,J)
ACAR(5,NB)=ACLNDZ
IACAF(1,NB)=IDPRKA(L)
2 IACAF(2,NB)=NSQ
DO 91 J=1,NSQ
NZ=NZ+1
91 FAFFCT(NZ) = APARSQ(L,J) / SPARSQ
1 CCNTINUE
C
DO 93 I=1,NLSEGS
93 NC(I)=0
NPKSRC=NB
DO 3 L=1,NPKSRC
DC 3 K=6,NT
HRACAR(K-5,L)=0.0
3 ACAR(K,L)=0.0
TVP= EXP(ALPHA(2)-BETA(2)/TEMK)
C
C BEGIN LOOP OVER N RUNWAYS
C
DC 10 N=1,NRNWYS
C
C IS RUNWAY USED WITH THIS WIND DIRECTION?
C
IF (IUSWD(IWD,N).EQ.0) GO TO 10
THETA=RNWY(7,N)
XO=0.25*DISRNW(N)*SIN(THETA)+RNWY(2,N)
YO=0.25*DISRNW(N)*COS(THETA)+RNWY(3,N)
NTT=NIBTT(N)
IF(NTT.EQ.0) GO TO 50
C
C BEGIN LOOP OVER J INBOUND TAXIWAYS

```

ACSRC062  
ACSRC063  
ACSPC064  
ACSRC065  
ACSRC066  
ACSRC067  
ACSPC068  
ACSRC069  
ACSRC070  
ACSRC071  
ACSPC072  
ACSRC073  
ACSRC074  
ACSRC075  
ACSRC076  
ACSRC077  
ACSRC078  
ACSRC079  
ACSRC080  
ACSRC081  
ACSRC082  
ACSPC083  
ACSRC084  
ACSRC085  
ACSRC086  
ACSRC087  
ACSRC088  
ACSRC089  
ACSRC090  
ACSRC091  
ACSRC092  
ACSRC093  
ACSRC094  
ACSRC095  
ACSRC096  
ACSRC097  
ACSRC098  
ACSRC099  
ACSRC100  
ACSPC101  
ACSRC102  
ACSRC103  
ACSRC104  
ACSRC105  
ACSRC106  
ACSRC107  
ACSRC108  
ACSRC109  
ACSRC110  
ACSRC111  
ACSRC112  
ACSRC113  
ACSRC114  
ACSRC115  
ACSRC116  
ACSRC117  
ACSRC118  
ACSRC119  
ACSRC120  
ACSRC121  
ACSRC122  
ACSRC123

```

C      DC 11 J=1,NTT          ACSRC124
C      ANY AIRCRAFT ARRIVING ON THIS RUNWAY?    ACSRC125
C
C      DC 7 I=1,NACTYP        ACSRC126
C      IF(TTARFR(J,I,N)*ARRFCN(23,I,N).GT.0.0) GO TO 701  ACSRC127
C      7 CCNTINUE            ACSPC128
C      GC TO 11              ACSRC129
C      701 NSGLNS = NIBSEG (J,N)          ACSPC130
C      BEGIN LOOP OVER K TAXIWAY SEGMENTS    ACSRC131
C
C      DC 12 K=1,NSGLNS        ACSRC132
C
C      SET UP SEGMENT LINE SOURCE GEOMETRIES  ACSPC133
C
C      JJ = IIBSEG(K,J,N)        ACSRC134
C      IF(NQ(JJ).NE.0) GO TO 130          ACSPC135
C      NC=NC+1                         ACSRC136
C      NQ(JJ)=NC                      ACSPC137
C      DO 121 L=1,12                 ACSRC138
C      121 ACLN(L,NC)=ACLNSG(L,JJ)      ACSPC139
C      ACLN(9,NC)=1.0                  ACSRC140
C      ACLN(10,NC)=1.0                 ACSPC141
C
C      ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS  ACSRC142
C      TO APPROPRIATE SEGMENTS          ACSPC143
C
C      DO 13 L=1,NPLTS            ACSRC144
C      LL=L+12                   ACSPC145
C      13  ACLN(LL,NC)=0.0        ACSRC146
C      130 ND=NQ(JJ)             ACSPC147
C      DC 14 I=1,NACTYP          ACSRC148
C      AA=ENGNO(I,1)             ACSPC149
C      IF(IEGFLG.GT.0) AA=ENGNO(I,2)      ACSRC150
C      ARR=TIARFR(J,I,N)*ARRFCN(23,I,N)*ANNARR(I)  ACSPC151
C      IF(ARR.LE.0.0) GO TO 14        ACSRC152
C      TIME=ACLN(11,ND)/(TXISED(I)*ACLN(9,JJ))    ACSPC153
C      FRC=AA*ARR*TIME*FRAC(I)      ACSRC154
C      DC 15 L=1,NPLTS          ACSPC155
C      KK=L+12                   ACSRC156
C      15  ACLN(KK,ND)=ACLN(KK,ND)+FRC*ACEMFC(I,2,L)  ACSPC157
C      14  CCNTINUE             ACSRC158
C      12  CONTINUE              ACSPC159
C
C      END TAXIWAY SEGMENT LOOP        ACSRC160
C
C      DETERMINE AIRCRAFT INBOUND PARKING AREA    ACSRC161
C      ASSOCIATED WITH TAXIWAY PATH          ACSPC162
C
C      DO 16 I=1,NPKSRC            ACSRC163
C      II=I                         ACSPC164
C      IDPK=IACAR(1,I)             ACSRC165
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 17        ACSPC166
C
C      16  CONTINUE                ACSRC167
C      PRINT 18, IDIEPA(J,N),J,N      ACSPC168
C
C      18  FORMAT ('INBOUND PARKING AREA ',I3,' OF TAXIWAY='I3,': RUNWAY='I3,'
C      1 IS NOT CONSISTANT WITH PARKING AREA ID NUMBERS')  ACSRC169
C      STOP                         ACSPC170
C
C      17  CONTINUE                ACSRC171
C
C
C      DO 19 I=1,NPKSRC            ACSPC172
C      II=I                         ACSRC173
C      IDPK=IACAR(1,I)             ACSPC174
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 17        ACSRC175
C
C      19  CONTINUE                ACSPC176
C      PRINT 19, IDIEPA(J,N),J,N      ACSRC177
C
C      20  FORMAT ('INBOUND PARKING AREA ',I3,' OF TAXIWAY='I3,': RUNWAY='I3,'
C      1 IS NOT CONSISTANT WITH PARKING AREA ID NUMBERS')  ACSPC178
C      STOP                         ACSRC179
C
C      21  CONTINUE                ACSPC180
C
C      DO 22 I=1,NPKSRC            ACSRC181
C      II=I                         ACSPC182
C      IDPK=IACAR(1,I)             ACSRC183
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 23        ACSPC184
C
C      22  CONTINUE                ACSRC185
C
C      DO 23 I=1,NPKSRC            ACSPC186
C      II=I                         ACSRC187
C      IDPK=IACAR(1,I)             ACSPC188
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 24        ACSRC189
C
C      23  CONTINUE                ACSPC190
C
C      DO 24 I=1,NPKSRC            ACSRC191
C      II=I                         ACSPC192
C      IDPK=IACAR(1,I)             ACSRC193
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 25        ACSPC194
C
C      24  CONTINUE                ACSRC195
C
C      DO 25 I=1,NPKSRC            ACSPC196
C      II=I                         ACSRC197
C      IDPK=IACAR(1,I)             ACSPC198
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 26        ACSRC199
C
C      25  CONTINUE                ACSPC200
C
C      DO 26 I=1,NPKSRC            ACSRC201
C      II=I                         ACSPC202
C      IDPK=IACAR(1,I)             ACSRC203
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 27        ACSPC204
C
C      26  CONTINUE                ACSRC205
C
C      DO 27 I=1,NPKSRC            ACSPC206
C      II=I                         ACSRC207
C      IDPK=IACAR(1,I)             ACSPC208
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 28        ACSRC209
C
C      27  CONTINUE                ACSPC210
C
C      DO 28 I=1,NPKSRC            ACSRC211
C      II=I                         ACSPC212
C      IDPK=IACAR(1,I)             ACSRC213
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 29        ACSPC214
C
C      28  CONTINUE                ACSRC215
C
C      DO 29 I=1,NPKSRC            ACSPC216
C      II=I                         ACSRC217
C      IDPK=IACAR(1,I)             ACSPC218
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 30        ACSRC219
C
C      29  CONTINUE                ACSPC220
C
C      DO 30 I=1,NPKSRC            ACSRC221
C      II=I                         ACSPC222
C      IDPK=IACAR(1,I)             ACSRC223
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 31        ACSPC224
C
C      30  CONTINUE                ACSRC225
C
C      DO 31 I=1,NPKSRC            ACSPC226
C      II=I                         ACSRC227
C      IDPK=IACAR(1,I)             ACSPC228
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 32        ACSRC229
C
C      31  CONTINUE                ACSPC230
C
C      DO 32 I=1,NPKSRC            ACSRC231
C      II=I                         ACSPC232
C      IDPK=IACAR(1,I)             ACSRC233
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 33        ACSPC234
C
C      32  CONTINUE                ACSRC235
C
C      DO 33 I=1,NPKSRC            ACSPC236
C      II=I                         ACSRC237
C      IDPK=IACAR(1,I)             ACSPC238
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 34        ACSRC239
C
C      33  CONTINUE                ACSPC240
C
C      DO 34 I=1,NPKSRC            ACSRC241
C      II=I                         ACSPC242
C      IDPK=IACAR(1,I)             ACSRC243
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 35        ACSPC244
C
C      34  CONTINUE                ACSRC245
C
C      DO 35 I=1,NPKSRC            ACSPC246
C      II=I                         ACSRC247
C      IDPK=IACAR(1,I)             ACSPC248
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 36        ACSRC249
C
C      35  CONTINUE                ACSPC250
C
C      DO 36 I=1,NPKSRC            ACSRC251
C      II=I                         ACSPC252
C      IDPK=IACAR(1,I)             ACSRC253
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 37        ACSPC254
C
C      36  CONTINUE                ACSRC255
C
C      DO 37 I=1,NPKSRC            ACSPC256
C      II=I                         ACSRC257
C      IDPK=IACAR(1,I)             ACSPC258
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 38        ACSRC259
C
C      37  CONTINUE                ACSPC260
C
C      DO 38 I=1,NPKSRC            ACSRC261
C      II=I                         ACSPC262
C      IDPK=IACAR(1,I)             ACSRC263
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 39        ACSPC264
C
C      38  CONTINUE                ACSRC265
C
C      DO 39 I=1,NPKSRC            ACSPC266
C      II=I                         ACSRC267
C      IDPK=IACAR(1,I)             ACSPC268
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 40        ACSRC269
C
C      39  CONTINUE                ACSPC270
C
C      DO 40 I=1,NPKSRC            ACSRC271
C      II=I                         ACSPC272
C      IDPK=IACAR(1,I)             ACSRC273
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 41        ACSPC274
C
C      40  CONTINUE                ACSRC275
C
C      DO 41 I=1,NPKSRC            ACSPC276
C      II=I                         ACSRC277
C      IDPK=IACAR(1,I)             ACSPC278
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 42        ACSRC279
C
C      41  CONTINUE                ACSPC280
C
C      DO 42 I=1,NPKSRC            ACSRC281
C      II=I                         ACSPC282
C      IDPK=IACAR(1,I)             ACSRC283
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 43        ACSPC284
C
C      42  CONTINUE                ACSRC285
C
C      DO 43 I=1,NPKSRC            ACSPC286
C      II=I                         ACSRC287
C      IDPK=IACAR(1,I)             ACSPC288
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 44        ACSRC289
C
C      43  CONTINUE                ACSPC290
C
C      DO 44 I=1,NPKSRC            ACSRC291
C      II=I                         ACSPC292
C      IDPK=IACAR(1,I)             ACSRC293
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 45        ACSPC294
C
C      44  CONTINUE                ACSRC295
C
C      DO 45 I=1,NPKSRC            ACSPC296
C      II=I                         ACSRC297
C      IDPK=IACAR(1,I)             ACSPC298
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 46        ACSRC299
C
C      45  CONTINUE                ACSPC300
C
C      DO 46 I=1,NPKSRC            ACSRC301
C      II=I                         ACSPC302
C      IDPK=IACAR(1,I)             ACSRC303
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 47        ACSPC304
C
C      46  CONTINUE                ACSRC305
C
C      DO 47 I=1,NPKSRC            ACSPC306
C      II=I                         ACSRC307
C      IDPK=IACAR(1,I)             ACSPC308
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 48        ACSRC309
C
C      47  CONTINUE                ACSPC310
C
C      DO 48 I=1,NPKSRC            ACSRC311
C      II=I                         ACSPC312
C      IDPK=IACAR(1,I)             ACSRC313
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 49        ACSPC314
C
C      48  CONTINUE                ACSRC315
C
C      DO 49 I=1,NPKSRC            ACSPC316
C      II=I                         ACSRC317
C      IDPK=IACAR(1,I)             ACSPC318
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 50        ACSRC319
C
C      49  CONTINUE                ACSPC320
C
C      DO 50 I=1,NPKSRC            ACSRC321
C      II=I                         ACSPC322
C      IDPK=IACAR(1,I)             ACSRC323
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 51        ACSPC324
C
C      50  CONTINUE                ACSRC325
C
C      DO 51 I=1,NPKSRC            ACSPC326
C      II=I                         ACSRC327
C      IDPK=IACAR(1,I)             ACSPC328
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 52        ACSRC329
C
C      51  CONTINUE                ACSPC330
C
C      DO 52 I=1,NPKSRC            ACSRC331
C      II=I                         ACSPC332
C      IDPK=IACAR(1,I)             ACSRC333
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 53        ACSPC334
C
C      52  CONTINUE                ACSRC335
C
C      DO 53 I=1,NPKSRC            ACSPC336
C      II=I                         ACSRC337
C      IDPK=IACAR(1,I)             ACSPC338
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 54        ACSRC339
C
C      53  CONTINUE                ACSPC340
C
C      DO 54 I=1,NPKSRC            ACSRC341
C      II=I                         ACSPC342
C      IDPK=IACAR(1,I)             ACSRC343
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 55        ACSPC344
C
C      54  CONTINUE                ACSRC345
C
C      DO 55 I=1,NPKSRC            ACSPC346
C      II=I                         ACSRC347
C      IDPK=IACAR(1,I)             ACSPC348
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 56        ACSRC349
C
C      55  CONTINUE                ACSPC350
C
C      DO 56 I=1,NPKSRC            ACSRC351
C      II=I                         ACSPC352
C      IDPK=IACAR(1,I)             ACSRC353
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 57        ACSPC354
C
C      56  CONTINUE                ACSRC355
C
C      DO 57 I=1,NPKSRC            ACSPC356
C      II=I                         ACSRC357
C      IDPK=IACAR(1,I)             ACSPC358
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 58        ACSRC359
C
C      57  CONTINUE                ACSPC360
C
C      DO 58 I=1,NPKSRC            ACSRC361
C      II=I                         ACSPC362
C      IDPK=IACAR(1,I)             ACSRC363
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 59        ACSPC364
C
C      58  CONTINUE                ACSRC365
C
C      DO 59 I=1,NPKSRC            ACSPC366
C      II=I                         ACSRC367
C      IDPK=IACAR(1,I)             ACSPC368
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 60        ACSRC369
C
C      59  CONTINUE                ACSPC370
C
C      DO 60 I=1,NPKSRC            ACSRC371
C      II=I                         ACSPC372
C      IDPK=IACAR(1,I)             ACSRC373
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 61        ACSPC374
C
C      60  CONTINUE                ACSRC375
C
C      DO 61 I=1,NPKSRC            ACSPC376
C      II=I                         ACSRC377
C      IDPK=IACAR(1,I)             ACSPC378
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 62        ACSRC379
C
C      61  CONTINUE                ACSPC380
C
C      DO 62 I=1,NPKSRC            ACSRC381
C      II=I                         ACSPC382
C      IDPK=IACAR(1,I)             ACSRC383
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 63        ACSPC384
C
C      62  CONTINUE                ACSRC385
C
C      DO 63 I=1,NPKSRC            ACSPC386
C      II=I                         ACSRC387
C      IDPK=IACAR(1,I)             ACSPC388
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 64        ACSRC389
C
C      63  CONTINUE                ACSPC390
C
C      DO 64 I=1,NPKSRC            ACSRC391
C      II=I                         ACSPC392
C      IDPK=IACAR(1,I)             ACSRC393
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 65        ACSPC394
C
C      64  CONTINUE                ACSRC395
C
C      DO 65 I=1,NPKSRC            ACSPC396
C      II=I                         ACSRC397
C      IDPK=IACAR(1,I)             ACSPC398
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 66        ACSRC399
C
C      65  CONTINUE                ACSPC400
C
C      DO 66 I=1,NPKSRC            ACSRC401
C      II=I                         ACSPC402
C      IDPK=IACAR(1,I)             ACSRC403
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 67        ACSPC404
C
C      66  CONTINUE                ACSRC405
C
C      DO 67 I=1,NPKSRC            ACSPC406
C      II=I                         ACSRC407
C      IDPK=IACAR(1,I)             ACSPC408
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 68        ACSRC409
C
C      67  CONTINUE                ACSPC410
C
C      DO 68 I=1,NPKSRC            ACSRC411
C      II=I                         ACSPC412
C      IDPK=IACAR(1,I)             ACSRC413
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 69        ACSPC414
C
C      68  CONTINUE                ACSPC415
C
C      DO 69 I=1,NPKSRC            ACSRC416
C      II=I                         ACSPC417
C      IDPK=IACAR(1,I)             ACSPC418
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 70        ACSRC419
C
C      69  CONTINUE                ACSPC420
C
C      DO 70 I=1,NPKSRC            ACSRC421
C      II=I                         ACSPC422
C      IDPK=IACAR(1,I)             ACSRC423
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 71        ACSPC424
C
C      70  CONTINUE                ACSRC425
C
C      DO 71 I=1,NPKSRC            ACSPC426
C      II=I                         ACSRC427
C      IDPK=IACAR(1,I)             ACSPC428
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 72        ACSRC429
C
C      71  CONTINUE                ACSPC430
C
C      DO 72 I=1,NPKSRC            ACSRC431
C      II=I                         ACSPC432
C      IDPK=IACAR(1,I)             ACSRC433
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 73        ACSPC434
C
C      72  CONTINUE                ACSRC435
C
C      DO 73 I=1,NPKSRC            ACSPC436
C      II=I                         ACSRC437
C      IDPK=IACAR(1,I)             ACSPC438
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 74        ACSRC439
C
C      73  CONTINUE                ACSPC440
C
C      DO 74 I=1,NPKSRC            ACSRC441
C      II=I                         ACSPC442
C      IDPK=IACAR(1,I)             ACSRC443
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 75        ACSPC444
C
C      74  CONTINUE                ACSRC445
C
C      DO 75 I=1,NPKSRC            ACSPC446
C      II=I                         ACSRC447
C      IDPK=IACAR(1,I)             ACSPC448
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 76        ACSRC449
C
C      75  CONTINUE                ACSPC450
C
C      DO 76 I=1,NPKSRC            ACSRC451
C      II=I                         ACSPC452
C      IDPK=IACAR(1,I)             ACSRC453
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 77        ACSPC454
C
C      76  CONTINUE                ACSRC455
C
C      DO 77 I=1,NPKSRC            ACSPC456
C      II=I                         ACSRC457
C      IDPK=IACAR(1,I)             ACSPC458
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 78        ACSRC459
C
C      77  CONTINUE                ACSPC460
C
C      DO 78 I=1,NPKSRC            ACSRC461
C      II=I                         ACSPC462
C      IDPK=IACAR(1,I)             ACSRC463
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 79        ACSPC464
C
C      78  CONTINUE                ACSRC465
C
C      DO 79 I=1,NPKSRC            ACSPC466
C      II=I                         ACSRC467
C      IDPK=IACAR(1,I)             ACSPC468
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 80        ACSRC469
C
C      79  CONTINUE                ACSPC470
C
C      DO 80 I=1,NPKSRC            ACSRC471
C      II=I                         ACSPC472
C      IDPK=IACAR(1,I)             ACSRC473
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 81        ACSPC474
C
C      80  CONTINUE                ACSRC475
C
C      DO 81 I=1,NPKSRC            ACSPC476
C      II=I                         ACSRC477
C      IDPK=IACAR(1,I)             ACSPC478
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 82        ACSRC479
C
C      81  CONTINUE                ACSPC480
C
C      DO 82 I=1,NPKSRC            ACSRC481
C      II=I                         ACSPC482
C      IDPK=IACAR(1,I)             ACSRC483
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 83        ACSPC484
C
C      82  CONTINUE                ACSRC485
C
C      DO 83 I=1,NPKSRC            ACSPC486
C      II=I                         ACSRC487
C      IDPK=IACAR(1,I)             ACSPC488
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 84        ACSRC489
C
C      83  CONTINUE                ACSPC490
C
C      DO 84 I=1,NPKSRC            ACSRC491
C      II=I                         ACSPC492
C      IDPK=IACAR(1,I)             ACSRC493
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 85        ACSPC494
C
C      84  CONTINUE                ACSRC495
C
C      DO 85 I=1,NPKSRC            ACSPC496
C      II=I                         ACSRC497
C      IDPK=IACAR(1,I)             ACSPC498
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 86        ACSRC499
C
C      85  CONTINUE                ACSPC500
C
C      DO 86 I=1,NPKSRC            ACSRC501
C      II=I                         ACSPC502
C      IDPK=IACAR(1,I)             ACSRC503
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 87        ACSPC504
C
C      86  CONTINUE                ACSRC505
C
C      DO 87 I=1,NPKSRC            ACSPC506
C      II=I                         ACSRC507
C      IDPK=IACAR(1,I)             ACSPC508
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 88        ACSRC509
C
C      87  CONTINUE                ACSPC510
C
C      DO 88 I=1,NPKSRC            ACSRC511
C      II=I                         ACSPC512
C      IDPK=IACAR(1,I)             ACSRC513
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 89        ACSPC514
C
C      88  CONTINUE                ACSRC515
C
C      DO 89 I=1,NPKSRC            ACSPC516
C      II=I                         ACSRC517
C      IDPK=IACAR(1,I)             ACSPC518
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 90        ACSRC519
C
C      89  CONTINUE                ACSPC520
C
C      DO 90 I=1,NPKSRC            ACSRC521
C      II=I                         ACSPC522
C      IDPK=IACAR(1,I)             ACSRC523
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 91        ACSPC524
C
C      90  CONTINUE                ACSRC525
C
C      DO 91 I=1,NPKSRC            ACSPC526
C      II=I                         ACSRC527
C      IDPK=IACAR(1,I)             ACSPC528
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 92        ACSRC529
C
C      91  CONTINUE                ACSPC530
C
C      DO 92 I=1,NPKSRC            ACSRC531
C      II=I                         ACSPC532
C      IDPK=IACAR(1,I)             ACSRC533
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 93        ACSPC534
C
C      92  CONTINUE                ACSRC535
C
C      DO 93 I=1,NPKSRC            ACSPC536
C      II=I                         ACSRC537
C      IDPK=IACAR(1,I)             ACSPC538
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 94        ACSRC539
C
C      93  CONTINUE                ACSPC540
C
C      DO 94 I=1,NPKSRC            ACSRC541
C      II=I                         ACSPC542
C      IDPK=IACAR(1,I)             ACSRC543
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 95        ACSPC544
C
C      94  CONTINUE                ACSRC545
C
C      DO 95 I=1,NPKSRC            ACSPC546
C      II=I                         ACSRC547
C      IDPK=IACAR(1,I)             ACSPC548
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 96        ACSRC549
C
C      95  CONTINUE                ACSPC550
C
C      DO 96 I=1,NPKSRC            ACSRC551
C      II=I                         ACSPC552
C      IDPK=IACAR(1,I)             ACSRC553
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 97        ACSPC554
C
C      96  CONTINUE                ACSRC555
C
C      DO 97 I=1,NPKSRC            ACSPC556
C      II=I                         ACSRC557
C      IDPK=IACAR(1,I)             ACSPC558
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 98        ACSRC559
C
C      97  CONTINUE                ACSPC560
C
C      DO 98 I=1,NPKSRC            ACSRC561
C      II=I                         ACSPC562
C      IDPK=IACAR(1,I)             ACSRC563
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 99        ACSPC564
C
C      98  CONTINUE                ACSRC565
C
C      DO 99 I=1,NPKSRC            ACSPC566
C      II=I                         ACSRC567
C      IDPK=IACAR(1,I)             ACSPC568
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 100        ACSRC569
C
C      99  CONTINUE                ACSPC570
C
C      DO 100 I=1,NPKSRC           ACSRC571
C      II=I                         ACSPC572
C      IDPK=IACAR(1,I)             ACSRC573
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 101        ACSPC574
C
C      100 CONTINUE                ACSRC575
C
C      DO 101 I=1,NPKSRC           ACSPC576
C      II=I                         ACSRC577
C      IDPK=IACAR(1,I)             ACSPC578
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 102        ACSRC579
C
C      101 CONTINUE                ACSPC580
C
C      DO 102 I=1,NPKSRC           ACSRC581
C      II=I                         ACSPC582
C      IDPK=IACAR(1,I)             ACSRC583
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 103        ACSPC584
C
C      102 CONTINUE                ACSPC585
C
C      DO 103 I=1,NPKSRC           ACSRC586
C      II=I                         ACSPC587
C      IDPK=IACAR(1,I)             ACSPC588
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 104        ACSRC589
C
C      103 CONTINUE                ACSPC590
C
C      DO 104 I=1,NPKSRC           ACSRC591
C      II=I                         ACSPC592
C      IDPK=IACAR(1,I)             ACSRC593
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 105        ACSPC594
C
C      104 CONTINUE                ACSPC595
C
C      DO 105 I=1,NPKSRC           ACSRC596
C      II=I                         ACSPC597
C      IDPK=IACAR(1,I)             ACSPC598
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 106        ACSRC599
C
C      105 CONTINUE                ACSPC600
C
C      DO 106 I=1,NPKSRC           ACSRC601
C      II=I                         ACSPC602
C      IDPK=IACAR(1,I)             ACSRC603
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 107        ACSPC604
C
C      106 CONTINUE                ACSPC605
C
C      DO 107 I=1,NPKSRC           ACSRC606
C      II=I                         ACSPC607
C      IDPK=IACAR(1,I)             ACSPC608
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 108        ACSRC609
C
C      107 CONTINUE                ACSPC610
C
C      DO 108 I=1,NPKSRC           ACSRC611
C      II=I                         ACSPC612
C      IDPK=IACAR(1,I)             ACSRC613
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 109        ACSPC614
C
C      108 CONTINUE                ACSPC615
C
C      DO 109 I=1,NPKSRC           ACSRC616
C      II=I                         ACSPC617
C      IDPK=IACAR(1,I)             ACSPC618
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 110        ACSRC619
C
C      109 CONTINUE                ACSPC620
C
C      DO 110 I=1,NPKSRC           ACSRC621
C      II=I                         ACSPC622
C      IDPK=IACAR(1,I)             ACSRC623
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 111        ACSPC624
C
C      110 CONTINUE                ACSPC625
C
C      DO 111 I=1,NPKSRC           ACSRC626
C      II=I                         ACSPC627
C      IDPK=IACAR(1,I)             ACSPC628
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 112        ACSRC629
C
C      111 CONTINUE                ACSPC630
C
C      DO 112 I=1,NPKSRC           ACSRC631
C      II=I                         ACSPC632
C      IDPK=IACAR(1,I)             ACSRC633
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 113        ACSPC634
C
C      112 CONTINUE                ACSPC635
C
C      DO 113 I=1,NPKSRC           ACSRC636
C      II=I                         ACSPC637
C      IDPK=IACAR(1,I)             ACSPC638
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 114        ACSRC639
C
C      113 CONTINUE                ACSPC640
C
C      DO 114 I=1,NPKSRC           ACSRC641
C      II=I                         ACSPC642
C      IDPK=IACAR(1,I)             ACSRC643
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 115        ACSPC644
C
C      114 CONTINUE                ACSPC645
C
C      DO 115 I=1,NPKSRC           ACSRC646
C      II=I                         ACSPC647
C      IDPK=IACAR(1,I)             ACSPC648
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 116        ACSRC649
C
C      115 CONTINUE                ACSPC650
C
C      DO 116 I=1,NPKSRC           ACSRC651
C      II=I                         ACSPC652
C      IDPK=IACAR(1,I)             ACSRC653
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 117        ACSPC654
C
C      116 CONTINUE                ACSPC655
C
C      DO 117 I=1,NPKSRC           ACSRC656
C      II=I                         ACSPC657
C      IDPK=IACAR(1,I)             ACSPC658
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 118        ACSRC659
C
C      117 CONTINUE                ACSPC660
C
C      DO 118 I=1,NPKSRC           ACSRC661
C      II=I                         ACSPC662
C      IDPK=IACAR(1,I)             ACSRC663
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 119        ACSPC664
C
C      118 CONTINUE                ACSPC665
C
C      DO 119 I=1,NPKSRC           ACSRC666
C      II=I                         ACSPC667
C      IDPK=IACAR(1,I)             ACSPC668
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 120        ACSRC669
C
C      119 CONTINUE                ACSPC670
C
C      DO 120 I=1,NPKSRC           ACSRC671
C      II=I                         ACSPC672
C      IDPK=IACAR(1,I)             ACSRC673
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 121        ACSPC674
C
C      120 CONTINUE                ACSPC675
C
C      DO 121 I=1,NPKSRC           ACSRC676
C      II=I                         ACSPC677
C      IDPK=IACAR(1,I)             ACSPC678
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 122        ACSRC679
C
C      121 CONTINUE                ACSPC680
C
C      DO 122 I=1,NPKSRC           ACSRC681
C      II=I                         ACSPC682
C      IDPK=IACAR(1,I)             ACSRC683
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 123        ACSPC684
C
C      122 CONTINUE                ACSPC685
C
C      DO 123 I=1,NPKSRC           ACSRC686
C      II=I                         ACSPC687
C      IDPK=IACAR(1,I)             ACSPC688
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 124        ACSRC689
C
C      123 CONTINUE                ACSPC690
C
C      DO 124 I=1,NPKSRC           ACSRC691
C      II=I                         ACSPC692
C      IDPK=IACAR(1,I)             ACSRC693
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 125        ACSPC694
C
C      124 CONTINUE                ACSPC695
C
C      DO 125 I=1,NPKSRC           ACSRC696
C      II=I                         ACSPC697
C      IDPK=IACAR(1,I)             ACSPC698
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 126        ACSRC699
C
C      125 CONTINUE                ACSPC700
C
C      DO 126 I=1,NPKSRC           ACSRC701
C      II=I                         ACSPC702
C      IDPK=IACAR(1,I)             ACSRC703
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 127        ACSPC704
C
C      126 CONTINUE                ACSPC705
C
C      DO 127 I=1,NPKSRC           ACSRC706
C      II=I                         ACSPC707
C      IDPK=IACAR(1,I)             ACSPC708
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 128        ACSRC709
C
C      127 CONTINUE                ACSPC710
C
C      DO 128 I=1,NPKSRC           ACSRC711
C      II=I                         ACSPC712
C      IDPK=IACAR(1,I)             ACSRC713
C      IF(IDPK.EQ.IDIBPA(J,N)) GO TO 129        ACSPC714
C
C      128 CONTINUE                ACSPC715
C
C      DO 129 I=1,NPKSRC           ACSRC716

```

```

C          ALLCCATE ALL AIRCRAFT IDLE AT SHUTDOWN, REFUELING,
C          AFRIVAL FUEL VENTING AND SERVICE VEHICLE EMISSIONS
C          TO APPROPRIATE AREA
C          NSQ=IACAR(2,II)
C          DC 19 I=1,NACTYP
C          ARR=TIARFR(J,I,N)*ARRFCN(23,I,N)*ANNARR(I)
C          IF(ARR.LE.0.0) GO TO 19
C          AA=ENGNO(I,1)
C          IF(IEGFLG.GT.0) AA=ENGNO(I,2)
C          TIME=SHIDNT(I)/60.
C          FFC=AA*ARR*TIME*FRAC(I)
C          TVP=EXP(ALPHA(JES1(I))-BETA(JES1(I))/TEMK)
C          DC 20 L=1,NSQ
C          JJ=II+L-1
C          DO 21 K=1,NPLTS
C          KK=K+5
C          ACAR(KK,JJ)=ACAR(KK,JJ)+FRC*ACEMPC(I,1,K)*PARFCT(JJ)
C          ACAR(KK,JJ)=ACAR(KK,JJ)+(AFSLEM(K,I,1)+ARSLEM(K,I,2) +
C          .ARSLEM(K,I,3)+ARSLEM(K,I,4)+ARSLEM(K,I,5))*ARR*FRAC(I)
C          .*PARFCT(JJ)
C          IF(K.EQ.2) ACAR(KK,JJ)=ACAR(KK,JJ)+(0.3*TVP*ACFUEL(I)*0.5
C          /1000.+ACSPIL(I)+ARFLVT(I))*ARR*FLDENS(JES1(I))*FRAC(I)
C          .*PARFCT(JJ)
21        CCNTINUE
20        CCNTINUE
19        CCNTINUE
11        CCNTINUE
C          END INBOUND TAXIWAY LOOP
C
C          BEGIN LOOP OVER I AIRCRAFT USED
C
C          DC 30 I=1,NACTYP
C
C          CALCULATE RUNWAY ARRIVALS FOR EACH AIRCRAFT TYPE
C
C          ARR=ARFFCN(23,I,N)*ANNARR(I)
C
C          ANY AIRCRAFT ARRIVING?
C
C          IF(ARR.LE.0.0) GO TO 30
C
C          SET UP LINE SOURCES FOR APPROACH AND LANDING MODES 7, 8 AND 9
C          AND ALLCCATE POLLUTANT EMISSIONS
C
C          AA=ENGNC(I,1)
C          DC 31 J=1,3
C          DO 32 K=1,3
C          KK=K+NC
C          JK=6*K-6+J
C          ACLN(J,KK)=ARFFCN(JK,I,N)
C          ACLN(J+5,KK)=ARRFCN(JK+6,I,N)
32        CCNTINUE
C          JJ=NC+J
C          JK=6*K-2
C          ACLN(4,JJ)=ARRFCN(24,I,N)
C          ACLN(5,JJ)=DEPFCN(24,I,N)
C          ACLN(09,JJ)=APRFCN(JK,I,N)
C          ACLN(10,JJ)=ARRFCN(JK+6,I,N)
          ACSRC186
          ACSRC187
          ACSRC188
          ACSRC189
          ACSRC190
          ACSRC191
          ACSRC192
          ACSRC193
          ACSRC194
          ACSRC195
          ACSRC196
          ACSRC197
          ACSRC198
          ACSRC199
          ACSRC200
          ACSRC201
          ACSRC202
          ACSRC203
          ACSRC204
          ACSRC205
          ACSRC206
          ACSRC207
          ACSRC208
          ACSRC209
          ACSRC210
          ACSRC211
          ACSRC212
          ACSRC213
          ACSRC214
          ACSRC215
          ACSRC216
          ACSRC217
          ACSRC218
          ACSRC219
          ACSRC220
          ACSRC221
          ACSRC222
          ACSRC223
          ACSRC224
          ACSRC225
          ACSRC226
          ACSRC227
          ACSRC228
          ACSRC229
          ACSRC230
          ACSRC231
          ACSRC232
          ACSRC233
          ACSRC234
          ACSRC235
          ACSRC236
          ACSRC237
          ACSRC238
          ACSRC239
          ACSRC240
          ACSRC241
          ACSRC242
          ACSRC243
          ACSRC244
          ACSRC245
          ACSRC246
          ACSRC247

```

```

ACLN(11,JJ)=ARRFCN(JK+1,I,N)          ACSRC248
ACLN(12,JJ)=AFRFCN(JK+2,I,N)          ACSRC249
JMODE=J+6                                ACSRC250
DO 33 R=1,NPLTS                         ACSRC251
KK=K+12                                 ACSRC252
ACLN(KK,JJ)=AA*ACEMFC(I,JMODE,K)*ARR*ARRFCN(JK+2,I,N)*FRAC(I) ACSRC253
33  CCNTINUE                                ACSRC254
31  CONTINUE                                 ACSRC255
NC=NC+3                                 ACSRC256
C
C      SET UP LINE SOURCES FOR TRAINING FLIGHT OPERATIONS
C      AND ALLOCATE POLLUTANT EMISSIONS
C
IF(ANNTGO(I).LE.0.0) GO TO 30          ACSRC257
NC=NC+1                                 ACSRC258
ACLN(1,NC)=XP(XO,TGO(1,1,I),THETA)    ACSRC259
ACLN(2,NC)=YP(YO,TGO(1,1,I),THETA)    ACSRC260
ACLN(6,NC)=XP(XO,TGO(1,2,I),THETA)    ACSRC261
ACLN(7,NC)=YP(YO,TGO(1,2,I),THETA)    ACSRC262
ACLN(1,NC+1)=ACLN(6,NC)                ACSRC263
ACLN(2,NC+1)=ACLN(7,NC)                ACSRC264
ACLN(6,NC+1)=XO                         ACSRC265
ACLN(7,NC+1)=YO                         ACSRC266
ACLN(1,NC+2)=XO                         ACSRC267
ACLN(2,NC+2)=YO                         ACSRC268
ACLN(6,NC+2)=XP(XO,0.3048,THETA)       ACSRC269
ACLN(7,NC+2)=YP(YO,0.3048,THETA)       ACSRC270
ACLN(1,NC+3)=ACLN(6,NC+2)              ACSRC271
ACLN(2,NC+3)=ACLN(7,NC+2)              ACSRC272
ACLN(6,NC+3)=XP(XO,TGO(1,3,I),THETA)  ACSRC273
ACLN(7,NC+3)=YP(YO,TGO(1,3,I),THETA)  ACSRC274
ACLN(1,NC+4)=ACLN(6,NC+3)              ACSRC275
ACLN(2,NC+4)=ACLN(7,NC+3)              ACSRC276
ACLN(6,NC+4)=XP(XO,TGO(1,4,I),THETA)  ACSRC277
ACLN(7,NC+4)=YP(YO,TGO(1,4,I),THETA)  ACSRC278
ACLN(3,NC)=APPHT*1000.                  ACSRC279
ACLN(8,NC)=APPHT2(I)*1000.              ACSRC280
ACLN(3,NC+1)=APPHT2(I)*1000.            ACSRC281
ACLN(8,NC+1)=ACLNDZ/2.                 ACSRC282
ACLN(3,NC+2)=ACLNDZ/2.                 ACSRC283
ACLN(8,NC+2)=ACLNDZ/2.                 ACSRC284
ACLN(3,NC+3)=ACLNDZ/2.                 ACSRC285
ACLN(8,NC+3)=COHT1(I)*1000.            ACSRC286
ACLN(3,NC+4)=COHT1(I)*1000.            ACSRC287
ACLN(8,NC+4)=CLMBHT*1000.              ACSRC288
ACLN(09,NC)=APSPD1(I)                  ACSRC289
ACLN(10,NC)=APSPD2(I)                  ACSRC290
ACLN(11,NC)=TGO(2,1,I)                 ACSRC291
ACLN(12,NC)=TGO(3,1,I)                 ACSRC292
ACLN(09,NC+1)=APSPD2(I)                ACSRC293
ACLN(10,NC+1)=LNDSPD(I)                ACSRC294
ACLN(11,NC+1)=TGO(2,2,I)                ACSRC295
ACLN(12,NC+1)=TGO(3,2,I)                ACSRC296
ACLN(09,NC+2)=LNDSPD(I)*1.3            ACSRC297
ACLN(10,NC+2)=TOSPD(I)*0.7             ACSRC298
ACLN(11,NC+2)=0.3048                  ACSRC299
ACLN(12,NC+2)=2.0*0.3048/(1.3*LNDSPD(I)+0.7*TOSPD(I)) ACSRC300
ACLN(09,NC+3)=TOSPD(I)                ACSRC301
ACLN(10,NC+3)=COSPD1(I)                ACSRC302
ACLN(11,NC+3)=TGO(2,3,I)                ACSRC303
ACLN(12,NC+3)=TGO(3,3,I)                ACSRC304
ACLN(09,NC+4)=COSPD1(I)                ACSRC305

```

```

ACLN(10,NC+4)=COSPD2(I)          ACSRC310
ACLN(11,NC+4)=TGO(2,4,I)          ACSRC311
ACLN(12,NC+4)=TGO(3,4,I)          ACSRC312
DO 45 J=1,5                      ACSRC313
JJ=NC+J-1                         ACSRC314
ACLN(4,JJ)=ARRFCN(24,I,N)         ACSRC315
ACLN(5,JJ)=DEPFCN(24,I,N)         ACSRC316
GO TO (34,35,41,36,37),J          ACSRC317
34  KD=7                           ACSRC318
  GC TO 38                         ACSRC319
35  KD=8                           ACSRC320
  GO TC 38                         ACSRC321
36  KD=5                           ACSRC322
  GO TO 38                         ACSRC323
37  KD=6                           ACSRC324
38  DO 39 K=1,NPLTS               ACSRC325
  KK=K+12                          ACSRC326
39  ACLN(KK,JJ)=ANNTGO(I)*ACEMFC(I,KD,K)*ARRFCN(23,I,N)*ACLN(12,JJ)*
  1FRAC(I)*AA                      ACSRC327
  GO TO 45                         ACSRC328
41  DO 42 K=1,NPLTS               ACSRC329
  KK=K+12                          ACSRC330
42  ACLN(KK,JJ)=AA*(0.3*ACEMFC(I,9,K)+0.7*ACEMFC(I,4,K))*1
  ANNIGC(I)*ARRFCN(23,I,N)*ACLN(12,JJ)*FRAC(I)          ACSRC331
45  CONTINUE                       ACSRC332
  NC=NC+4                          ACSRC333
30  CONTINUE                       ACSRC334
C   END AIRCRAFT LOOP             ACSRC335
C
C   50  NTT=NOBTT(N)               ACSRC336
  IF(NTT.EQ.0) GO TO 10            ACSRC337
C
C   BEGIN LOOP OVER J OUTBOUND TAXIWAYS
C
C   DC 51 J=1,NTT                  ACSRC338
C
C   ANY AIRCRAFT DEPARTING ON THIS TAXIWAY?
C
C   DC 6 I=1,NACTYP                ACSRC339
  IF(TTDPFR(J,I,N)*DEPFCN(23,I,N).GT.0.0) GO TO 601
6  CONTINUE                         ACSRC340
  GO TO 51                          ACSRC341
601 NSGLNS=NOBSEG(J,N)            ACSRC342
C
C   BEGIN LLOOP OVER K TAXIWAY SEGMENTS
C
C   DC 52 K=1,NSGLNS               ACSRC343
C
C   SET UP SEGMENT LINE SOURCE GEOMETRIES
C
C   JJ=IOBSEG(K,J,N)               ACSRC344
  IF(NC(JJ).NE.0) GO TO 131
  NC=NC+1                          ACSRC345
  NQ(JJ)=NC                         ACSRC346
  DC 122 L=1,12                     ACSRC347
122 ACLN(L,NC)=ACLNNG(L,JJ)       ACSRC348
  ACLN(9,NC)=1.0                     ACSRC349
  ACLN(10,NC)=1.0                    ACSRC350
C
C   ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS
C   TO APPROPRIATE SEGMENTS          ACSRC351
C

```

```

C
      DC 53 L=1,NPLTS          ACSRC372
      LL=L+12                  ACSRC373
  53  ACLN(LL,NC)=0.0          ACSRC374
  131 ND=NC(JJ)               ACSRC375
      DC 54 I=1,NACTYP          ACSRC376
      DEP=TTDPFR(J,I,N)*DEPFCN(23,I,N)*ANNDEP(I)  ACSRC377
      IF(DEP.LE.0.0) GO TO 54   ACSRC378
      AA=ENGNO(I,1)             ACSPC379
      IF(IEGFLG.GT.0) AA=ENGNO(I,2)  ACSRC380
      TIME= ACLN(11,ND) / TXISPD(I)  ACSRC381
      FRC= AA* DEP*TIME*FRAC(I)    ACSRC382
      DO 55 L=1,NPLTS          ACSRC383
      KK=L+12                  ACSRC384
  55  ACLN(KK,ND)= ACLN(KK,ND)+FRC*ACEMFC(I,2,L)  ACSRC385
  54  CCNTINUE                ACSRC386
  52  CCNTINUE                ACSRC387
C
C      END TAXIWAY SEGMENT LOOP  ACSRC388
C
C
C      DETERMINE AIRCRAFT OUTBOUND PARKING AREA ASSOCIATED  ACSRC389
C      WITH TAXIWAY PATH          ACSRC390
C
C      DO 56 I=1,NPKSRC          ACSRC391
      II =I                      ACSRC392
      IDPK=IACAR(1,I)            ACSRC393
      IF(IDPK.EQ.IDOOPA(J,N)) GO TO 58  ACSRC394
  56  CONTINUE                  ACSRC395
      PRINT 57,IDOOPA(J,N),J,N      ACSRC396
  57  FORMAT(22H0CUTBOUND PARKING AREA,I3,11H OF TAXIWAY,I3,8H, RUNWAY,  ACSRC400
      . I3,47H IS NOT CONSISTENT WITH PARKING AREA ID NUMBERS)  ACSRC401
      STOP                      ACSRC402
C
C      ALLOCATE ALL AIRCRAFT IDLE AT STARTUP, DEPARTURE FUEL  ACSRC403
C      VENTING AND SERVICE VEHICLE EMISSIONS TO APPROPRIATE AREA  ACSRC404
C
  58  NSQ=IACAR(2,II)
      DO 59 I=1,NACTYP          ACSRC405
      DEP=TTDPFR(J,I,N)*DEPFCN(23,I,N)*ANNDEP(I)  ACSRC406
      IF(DEP.EQ.0.0) GO TO 59  ACSRC407
      AA=ENGNO(I,1)             ACSRC408
      IF(IEGFLG.GT.0) AA=ENGNO(I,2)  ACSRC409
      TIME=SRTUPT(I)/60.          ACSRC410
      FRC = AA* DEP* TIME * FRAC(I)  ACSRC411
      TVP=EXP(ALPHA(JES1(I)) - BETA(JES1(I)) / TEMK)  ACSRC412
      DC 60 L=1,NSQ              ACSRC413
      JJ =II +L-1                ACSRC414
      DO 61 K=1,NPLTS          ACSRC415
      KK=K+5
      ACAR(KK,JJ) = ACAR(KK,JJ) + ((FRC * ACEMFC(I,1,K)) +  ACSRC416
      . ((DFSSEM(K,I,1) + DPSSEM(K,I,2) + DPSSEM(K,I,3) + DPSSEM(K,I,4)  ACSRC417
      . + DPSSEM(K,I,5)) * DEP * FRAC(I)) * PARFCT(JJ)  ACSRC418
      IF (K.EQ.2) ACAR(KK,JJ) = ACAR(KK,JJ) + DPFLVT(I) * DEP * FLDENS(  ACSRC419
      . JES1(I)) * FRAC(I) * PARFCT(JJ)  ACSRC420
  61  CCNTINUE                ACSRC421
  60  CCNTINUE                ACSRC422
  59  CCNTINUE                ACSRC423
  51  CCNTINUE                ACSRC424
C
C      END OUTBOUND TAXIWAY LOOP  ACSRC425
C

```

```

C NB=NB+1
C SET UP AREA SOURCE AT TAIL OF RUNWAY AND ALLOCATE
C ENGINE CHECK EMISSIONS TO IT
C
C ACAR (1,NB)=RNWY (2,N) -.05 * SIN (THETA)
C ACAR (2,NB)=RNWY (3,N) -.05*COS (THETA)
C ACAR (3,NB)= ACLNDZ/2.
C ACAR (4,NB)= 100.0
C ACAR (5,NB)= ACLNDZ
C DO 65 K=1,NPLTS
C KK=K+5
65 ACAR (KK,NB)=0.0
C DO 66 I=1,NACTYP
C DEP=DEPFCN (23,I,N)*ANNDEP (I)
C IF (DEP.EQ.0.0) GO TO 66
C AA=ENGNO (I,1)
C IF (IEGFLG.GT.0) AA=ENGNO (I,2)
C TIME= EGCHKT (I)/60.
C FRC= TIME *DEP*AA*FRAC (I)
C DC 67 K=1,NELTS
C KK=K+5
67 ACAR (KK,NB)=ACAR (KK,NB) + FRC* ACEMPC (I,3,K)
66 CONTINUE
10 CONTINUE
C
C END RUNWAY LOOP
C
C NACAR=NB
C NC1=NC
C NC=NC1
69 BEGIN LOOP OVER N RUNWAYS
C
C DO 79 N=1,NRNWYS
C
C IS RUNWAY USED WITH THIS WIND DIRECTION?
C
C IF (IUSWD (IWD,N).EQ.0) GO TO 79
C
C BEGIN LOOP OVER I AIRCRAFT USED
C
C DO 70 I=1,NACTYP
C
C CALCULATE RUNWAY DEPARTURES FOR EACH AIRCRAFT TYPE
C
C DEP=DEPFCN (23,I,N)*ANNDEP (I)
C
C ANY AIRCRAFT DEPARTING FROM THIS RUNWAY?
C
C IF (DEP.EQ.0.0) GO TO 70
C
C CALL DEPART TO CALCULATE POINTS IN TAKEOFF PATH ACCORDING
C TO CURRENT METEOROLOGICAL CONDITIONS
C
C CALL DEPART (N,I)
C AA=ENGNO (I,1)
C
C SET UP LINE SOURCES FOR RUNWAY ROLL AND CLIMBOUT MODES 1 AND 2
C AND ALLOCATE POLLUTANT EMISSIONS
C
C DO 71 J=1,3

```

DO 72 K=1,3	ACSRC496
KK=K+NC	ACSRC497
JK=6*K-6+J	ACSRC498
ACLN (J, KK) =DEPPCN (JK, I, N)	ACSPC499
72 ACLN (J+5, KK) =DEPPCN (JK+6, I, N)	ACSRC500
JJ=NC+J	ACSRC501
JK=6*K-2	ACSRC502
ACLN (4, JJ) =ARRFCN (24, I, N)	ACSRC503
ACLN (5, JJ) =DEPPCN (24, I, N)	ACSRC504
ACLN (09, JJ) =DEPPCN (JK, I, N)	ACSRC505
ACLN (10, JJ) =DEPPCN (JK+6, I, N)	ACSRC506
ACLN (11, JJ) =DEPPCN (JK+1, I, N)	ACSRC507
ACLN (12, JJ) =DEPPCN (JK+2, I, N)	ACSRC508
JMODE=J+3	ACSRC509
DO 73 K=1,NPLTS	ACSRC510
KK=K+12	ACSRC511
ACLN (KK, JJ) =AA*ACEMFC (I, JMODE, K) *D&P*DEPPCN (JK+2, I, N) *FRAC (I)	ACSRC512
73 CONTINUE	ACSRC513
71 CONTINUE	ACSRC514
NC=NC+3	ACSRC515
70 CCNTINUE	ACSRC516
C END AIRCRAFT LOOP	ACSRC517
C	ACSRC518
C 79 CCNTINUE	ACSRC519
C	ACSRC520
C END RUNWAY LOOP	ACSRC521
C	ACSRC522
NACLN=NC	ACSRC523
RETURN	ACSPC524
END	ACSPC525
	ACSRC526

## FUNCTION AINE

### Purpose:

1. To translate the line and receptor coordinates to an x-axis along the wind vector, placing the origin of the line at its low end.
2. To set up the necessary parameters for the CAVL and QMOD routines.
3. To determine the concentration due to the given line.

### Input:

The current wind direction and speed, and the receptor and line source data.

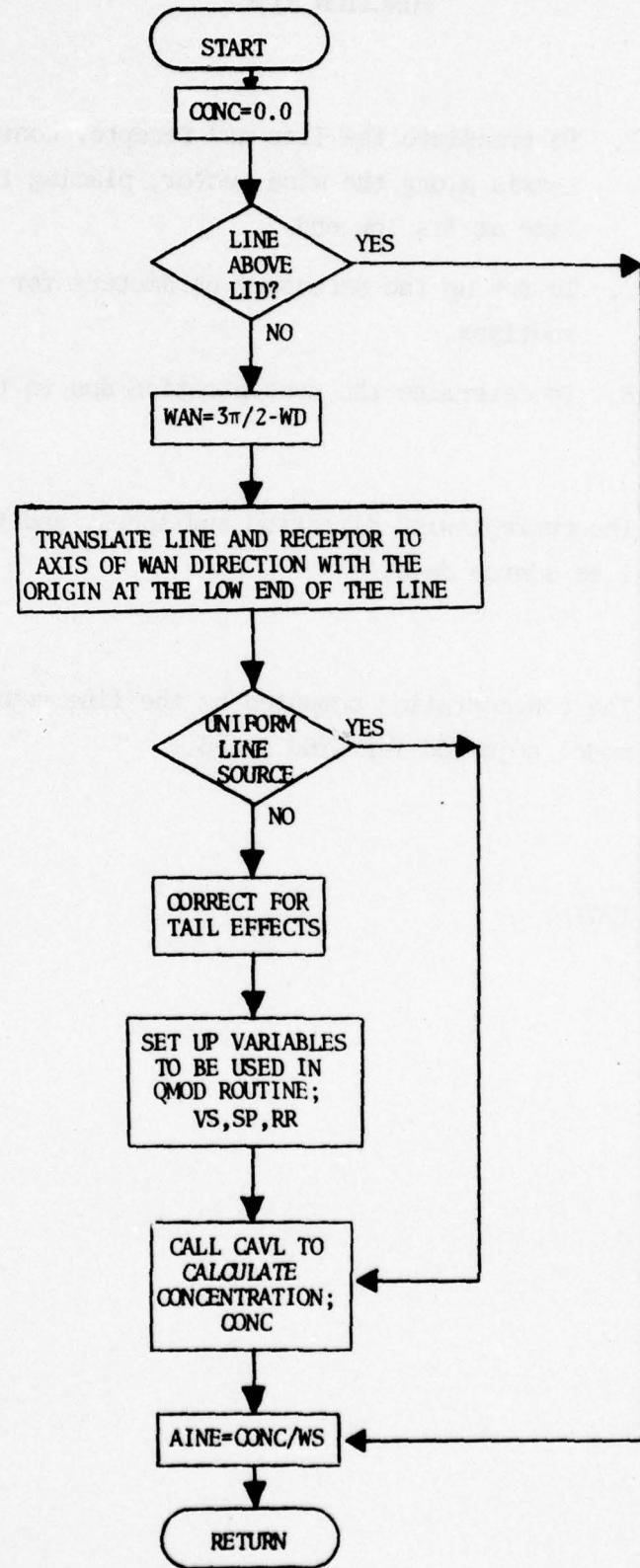
### Output:

The concentration computed by the line source diffusion model adjusted for wind speed.

### Subroutines Called:

CAVL

FUNCTION AINE(WD)



```

C      FUNCTION AINE (WD)          AINE0000
C
C      THIS FUNCTION TRANSLATES THE LINE AND RECEPTOR COORDINATES TO AN  AINE0001
C      X-AXIS ALONG THE WIND VECTOR, PLACING THE ORIGIN OF THE LINE AT  AINE0002
C      ITS LOW END.  THE VEHICLE MOVES FROM (X1,Y1,Z1) TO (X2,Y2,Z2)  AINE0003
C
C      COMMON /MFT/ WS,WSMPH,IWS,WX,IWD,SINNEWD,COSEWD,JSTAB,HLID,TEMP,  AINE0004
C      . TEMK
C      COMMON /RCPT/ NRECEP,RECEP(2,312)  AINE0005
C      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,AINE0006
C      . V1,V2,DL,TIME,EMIS(6),NPOL  AINE0007
C      COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,AINE0008
C      . WS2,WSC,RR,SP,XST,YST,ZST,XND,YND,ZND  AINE0009
C      DATA PI32/4.7123890/  AINE0010
C
C      CCNC = 0.  AINE0011
C
C      IF LINE IS ABOVE LID, DO NOT CALCULATE CONC  AINE0012
C
C      IF(ZW1.GE.HLID-.5) GO TO 60  AINE0013
C
C      TRANSLATE LINE AND RECEPTOR TO AXIS OF WAN DIRECTION  AINE0014
C
C      WAN=PI32-WD  AINE0015
C      CSAN=COS(WAN)  AINE0016
C      SNAN=STN(WAN)  AINE0017
C      XW2=(X2-X1)*CSAN+(Y2-Y1)*SNAN  AINE0018
C      YW2=(X1-X2)*SNAN+(Y2-Y1)*CSAN  AINE0019
C      XR = RECEP(1,IRECEP) * 1000.  AINE0020
C      YE = PECEP(2,IRECEP) * 1000.  AINE0021
C      ZST=ZW1  AINE0022
C      ZND=ZW2  AINE0023
C      IF(Z1.LE.Z2) GO TO 5  AINE0024
C      XW2=-XW2  AINE0025
C      XST=XW2  AINE0026
C      YW2=-YW2  AINE0027
C      YST=YW2  AINE0028
C      XND=0.0  AINE0029
C      YND=0.0  AINE0030
C      XRCP=(XR-X2)*CSAN+(YR-Y2)*SNAN  AINE0031
C      YRCP=(X2-XR)*SNAN+(YR-Y2)*CSAN  AINE0032
C      GO TO 8  AINE0033
C
5  CCNTINUE  AINE0034
C      XST=0.0  AINE0035
C      YST=0.0  AINE0036
C      XND=XW2  AINE0037
C      YND=YW2  AINE0038
C      XRCP=(XR-X1)*CSAN+(YR-Y1)*SNAN  AINE0039
C      YRCP=(X1-XR)*SNAN+(YR-Y1)*CSAN  AINE0040
C      GO TO 8  AINE0041
C
5  CCNTINUE  AINE0042
C      XST=0.0  AINE0043
C      YST=0.0  AINE0044
C      XND=XW2  AINE0045
C      YND=YW2  AINE0046
C      XRCP=(XR-X1)*CSAN+(YR-Y1)*SNAN  AINE0047
C      YRCP=(X1-XR)*SNAN+(YR-Y1)*CSAN  AINE0048
C
8  CONTINUE  AINE0049
C      ZRCP = 2.  AINE0050
C
C      IS THIS A UNIFORM LINE SOURCE  AINE0051
C
C      50 IF(IAD.EQ.0) GO TO 500  AINE0052
C
C      CORRECT FOR TAIL EFFECTS IF ARRIVAL OR DEPARTURE  AINE0053
C
C      CSA = -XW2 / DL  AINE0054
C      WSC = 2 * WS * CSA  AINE0055
C      EXT = TAIL / DL  AINE0056
C      DX = XW2 * EXT  AINE0057
C
C      AINE0058
C      AINE0059
C      AINE0060
C      AINE0061

```

DX = YW2 * EXT	AINE0062
XW2 = XW2 + DX	AINE0063
YW2 = YW2 + DY	AINE0064
VS = TAIL / TIME	AINE0065
W1 = V1 + VS	AINE0066
W2 = V2 + VS	AINE0067
YY1 = SQRT(WS2 + W1 * (W1 + WSC))	AINE0068
YY2 = SQRT(WS2 + W2 * (W2 + WSC))	AINE0069
SP = YY2	AINE0070
ARG = (YY2 + W2 + WSC/2.) / (YY1 + W1 + WSC/2.)	AINE0071
G = YY2 - YY1 - WSC/2. * ALOG(ARG)	AINE0072
RR = A / G	AINE0073
IF (Z1.NE.Z2.AND.IAD.EQ.1) GO TO 500	AINE0074
XRCP = XRCP + DX	AINE0075
YRCP = YRCP + DY	AINE0076
C	AINE0077
C    CALCULATE THE CONCENTRATION DUE TO THIS LINE	AINE0078
C	AINE0079
500 CONC=CAVL(XRCP,YRCP,ZRCP)	AINE0080
60 AINE = CONC / WS	AINE0081
RETURN	AINE0082
END	AINE0083

## BLOCK DATA

Purpose:

To initialize data in common blocks.

Input:

None

Output:

None

```

      BLOCK DATA                                BLKDT000
C                                             BLKDT001
C   INITIALIZE DATA IN COMMON BLOCKS FOR SHORT TERM MODEL   BLKDT002
C                                             BLKDT003
C   REAL*8 POLNAM                                BLKDT004
C                                             BLKDT005
C
C   COMMON /ANNMET/ TBAR,ADD,PA,WSBAR,DTBAR      BLKDT006
C   COMMON /DEFALT/ ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)  BLKDT007
C   COMMON /DSTRET/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),  BLKDT008
.  VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),  BLKDT009
.  CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1  BLKDT010
C   COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,B,V12,VS,  BLKDT011
.  WS2,WSC,RR,SP,AA1,AA2,AA3,AA4,AA5,AA6      BLKDT012
COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG,IONCE  BLKDT013
COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,  BLKDT014
.  NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),  BLKDT015
.  ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)  BLKDT016
COMMON /TITL/ POLNAM( 6),TITLE1(20),IPCHOS( 6),NXPOL,IF  BLKDT017
C   COMMON /WNDPRO/ XP(6)                      BLKDT018
C                                             BLKDT019
C   *****DATA STATEMENTS*****                  BLKDT020
C                                             BLKDT021
DATA XW1, YW1, TAIL / 0.0, 0.0, 140. /      BLKDT022
DATA XP / 0.2, 0.2, 0.2, 0.3, 0.4, 0.4 /      BLKDT023
DATA ALPHA / 11.70365, 11.10675, 12.42382, 12.68789, 13.687,  BLKDT024
. 13.038, 13.024 /      BLKDT025
DATA BETA / 2868.54, 3129.5187, 3276.8848, 5108.4194, 5329.139,  BLKDT026
. 4789.301, 4782.209 /      BLKDT027
DATA FLDENS / 0.695, 0.773, 0.693, 0.842, 0.824, 0.807, 0.807 /  BLKDT028
DATA ACLNDY, ACLNDZ / 20.0,8.0 /      BLKDT029
DATA ITAPE, IONCE / 21, 0 /      BLKDT030
DATA POLNAM / 8H CO ,8H HC ,8H NOX ,8H PT ,  BLKDT031
. 8H SO2 ,8H POL6 /      BLKDT032
DATA ENPT,ENAR,ENLN,ABPT,ABLN,ACPT,ACAR,ACLN / 12660*0.0 /  BLKDT033
DATA NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,NACAR,NACLN /9*0.0/BLKDT034
END                                         BLKDT035

```

## FUNCTION CAVL

### Purpose:

To compute the coupling coefficient at a receptor due to a line source.

### Input:

Meteorological conditions: wind speed; stability; mixing height; critical distance for vertical mixing; psuedo downwind distances for horizontal and vertical spreads of the line source.

Source parameters: end point coordinates of the line (X-axis has been chosen to be along the wind vector); IAD flag for uniform or non-uniform line.

Receptor coordinates.

### Output:

CAVL, the coupling coefficient.

### Procedure:

1. Test whether the receptor is located with respect to the line source such that the concentration is negligible.
2. If the angle between the wind vector and line is sufficiently small, and the line is sufficiently long, set a flag for the line to be segmented.
3. Compute effective downwind distance and the horizontal and vertical dispersion coefficients.
4. Determine factor to be used in subdividing the line.
5. Test whether the line has a uniform density. If it is a runway used for aircraft arrival or departure (non-uniform density), call subroutine QMOD.
6. Determine the proper expression to be used and compute the concentration due to the line segment.
7. Test whether further segments need be considered. If not, output the concentration for the given receptor.

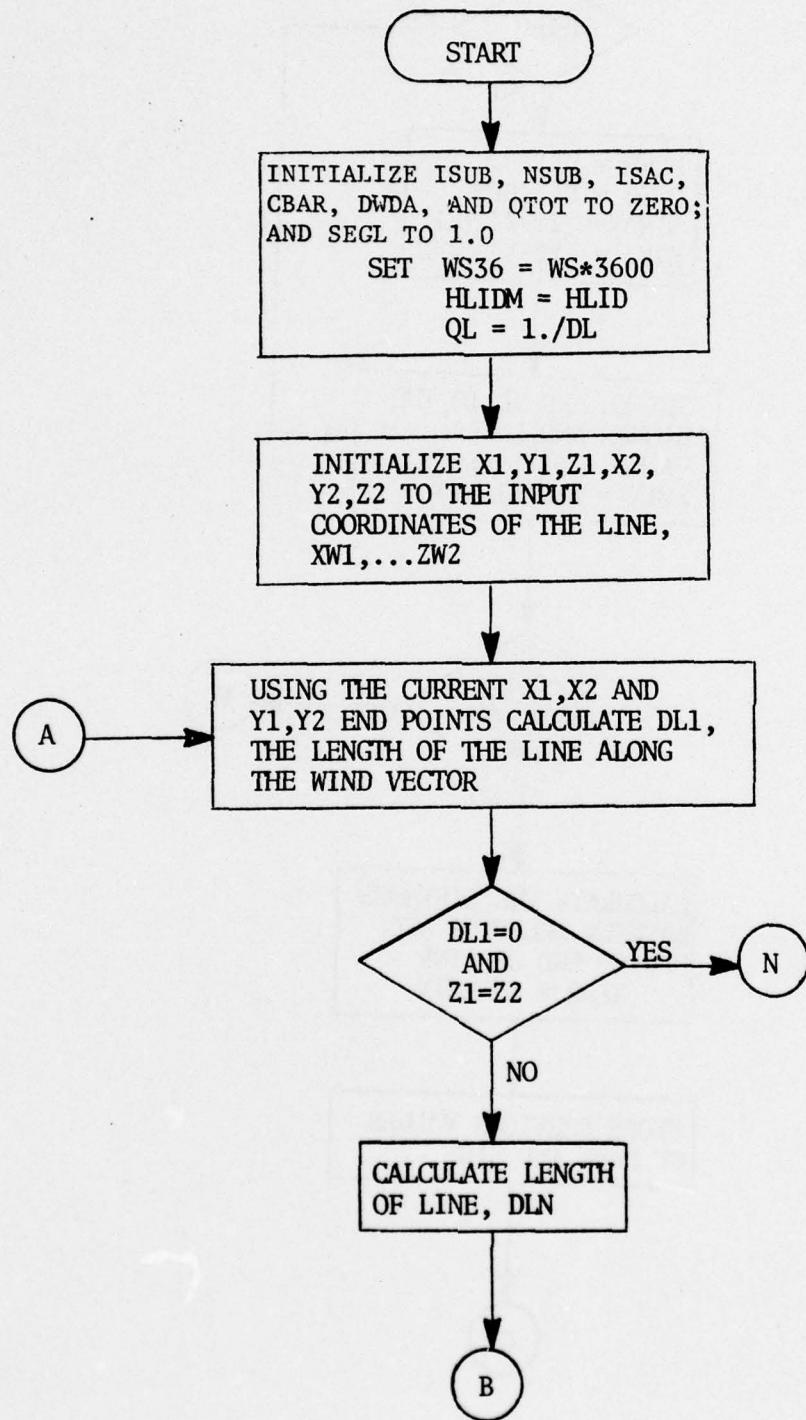
**Functions  
Called:**

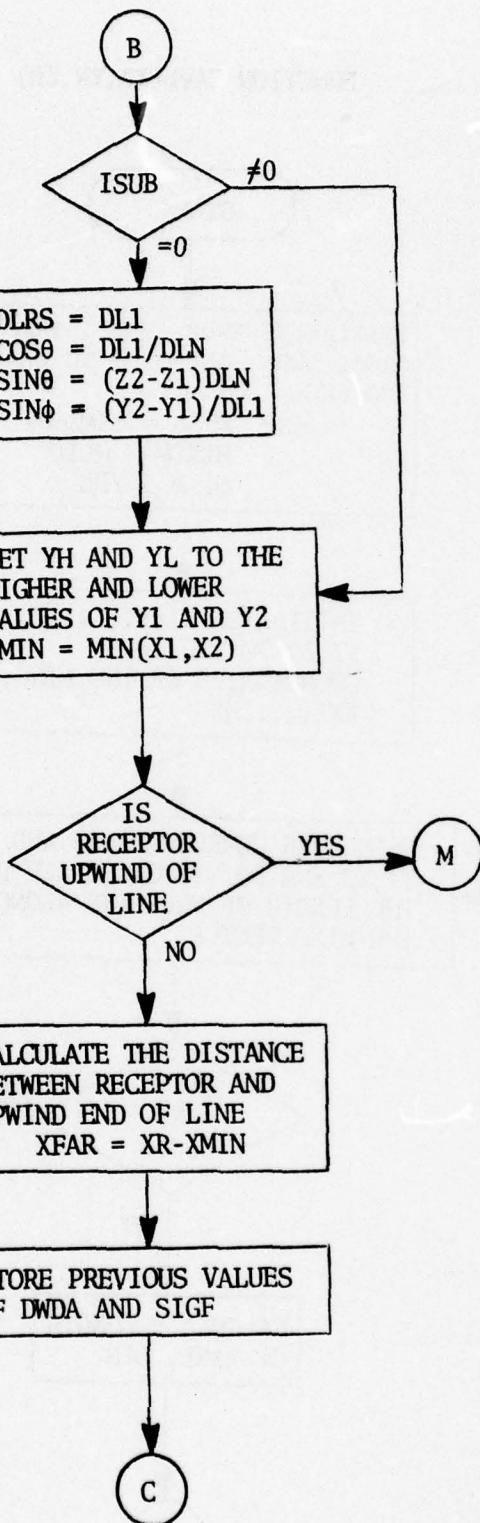
SIGY, SIGZ, DIFERF

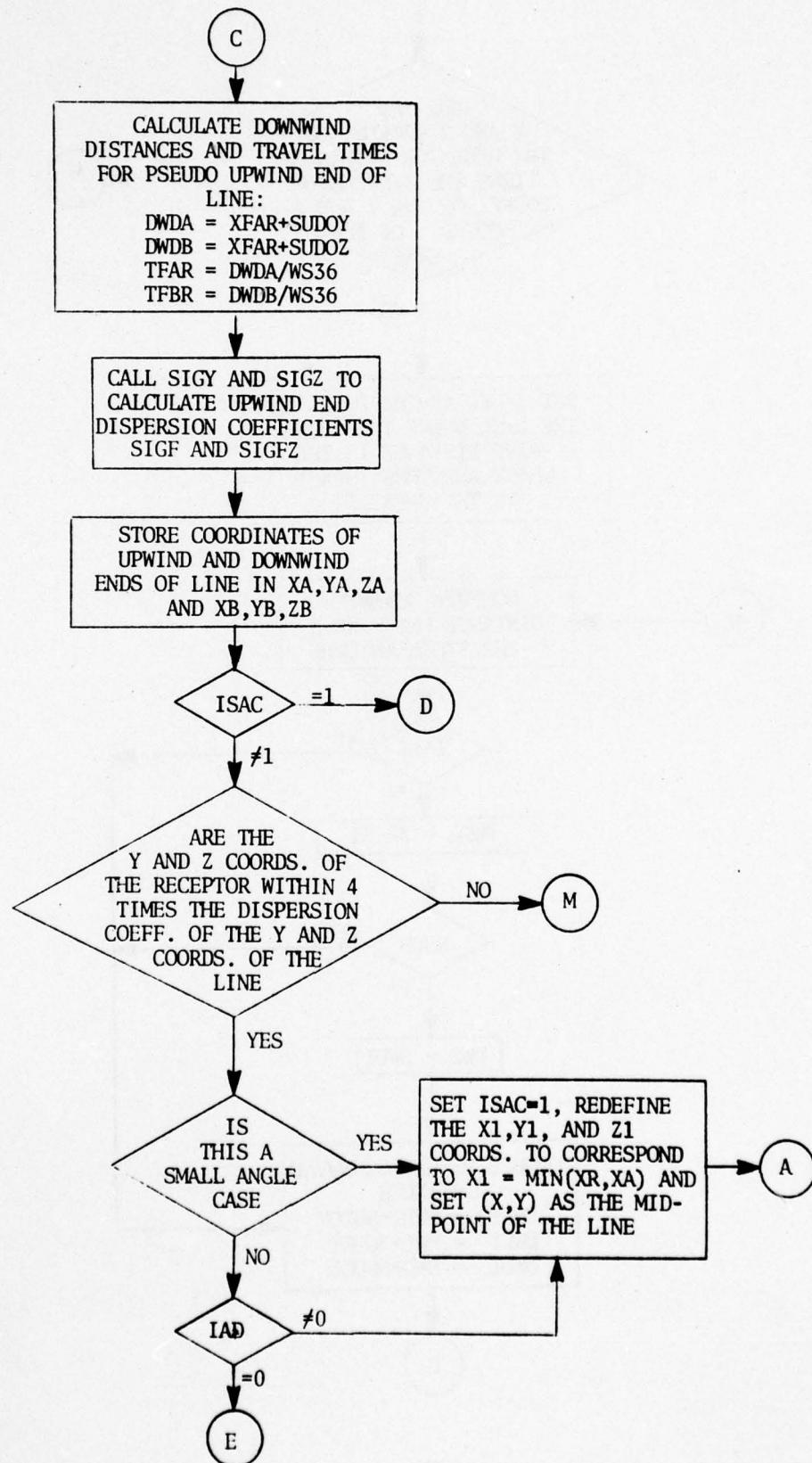
**Subroutine  
Called:**

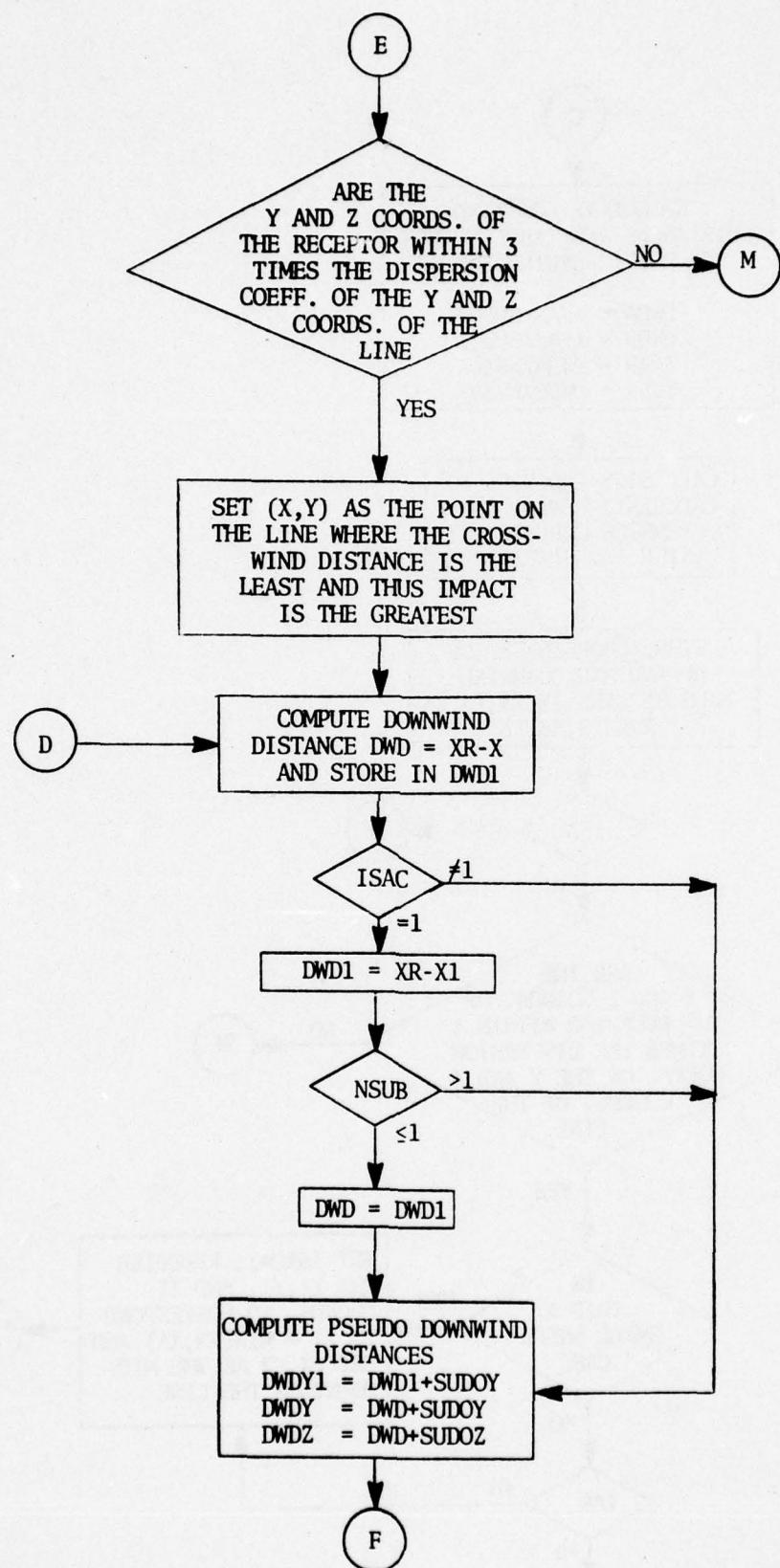
QMOD

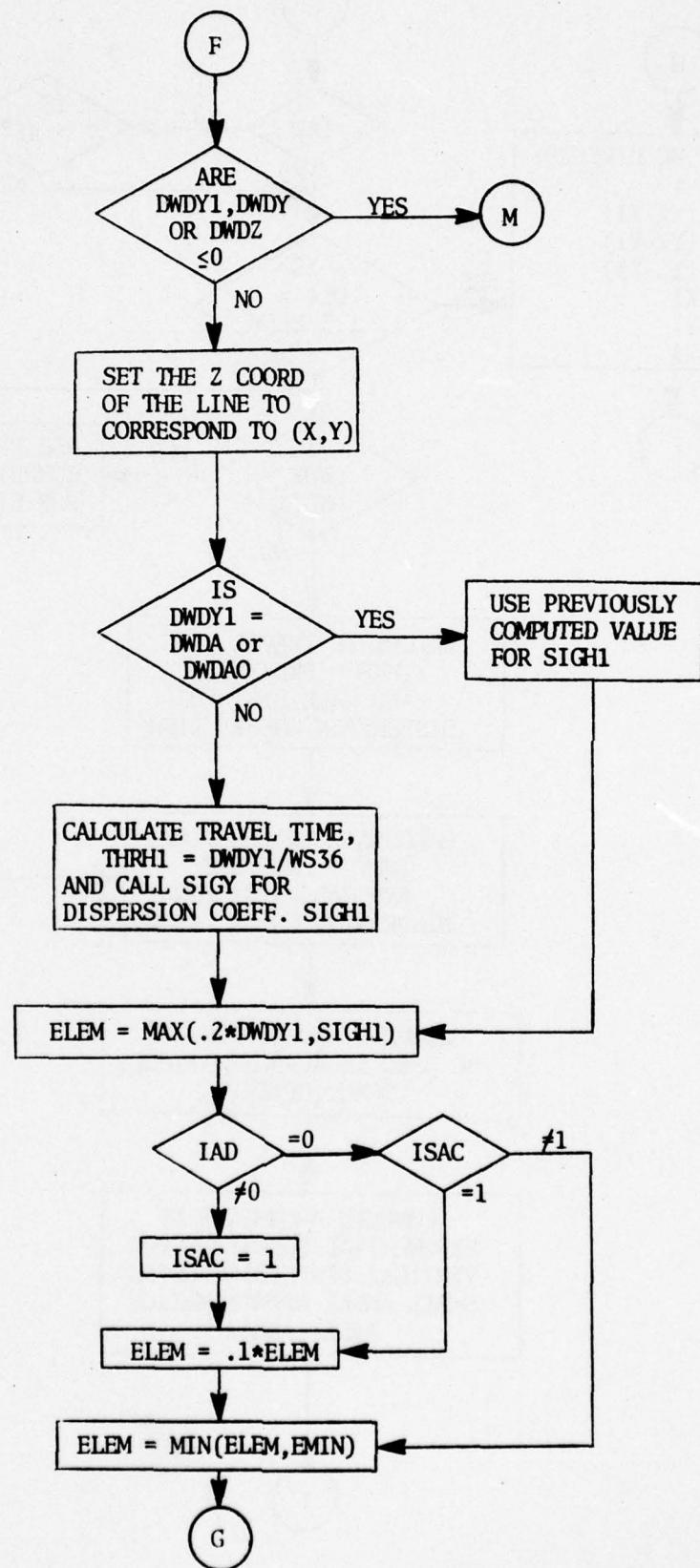
FUNCTION CAVL(XR,YR,ZR)

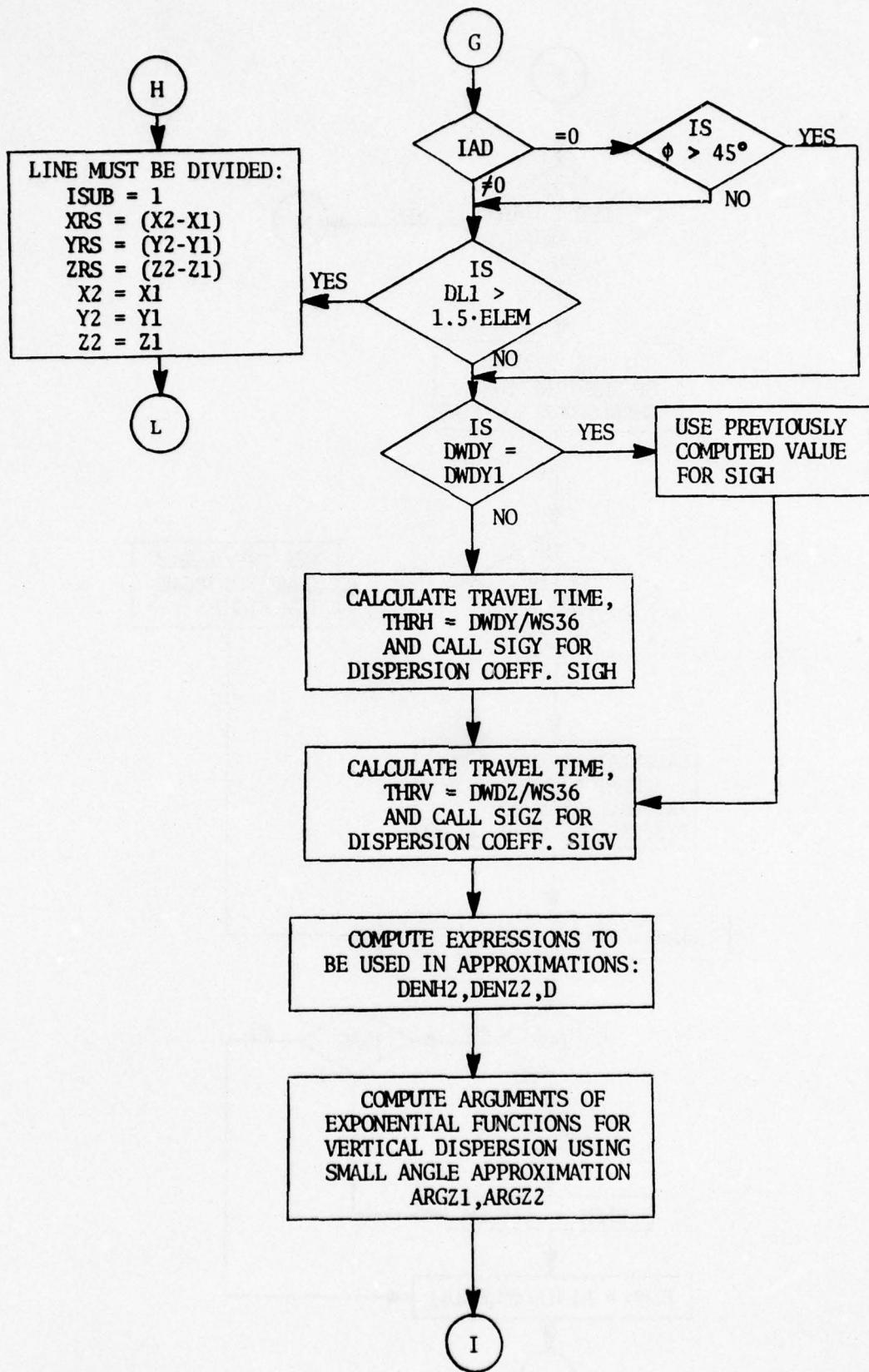


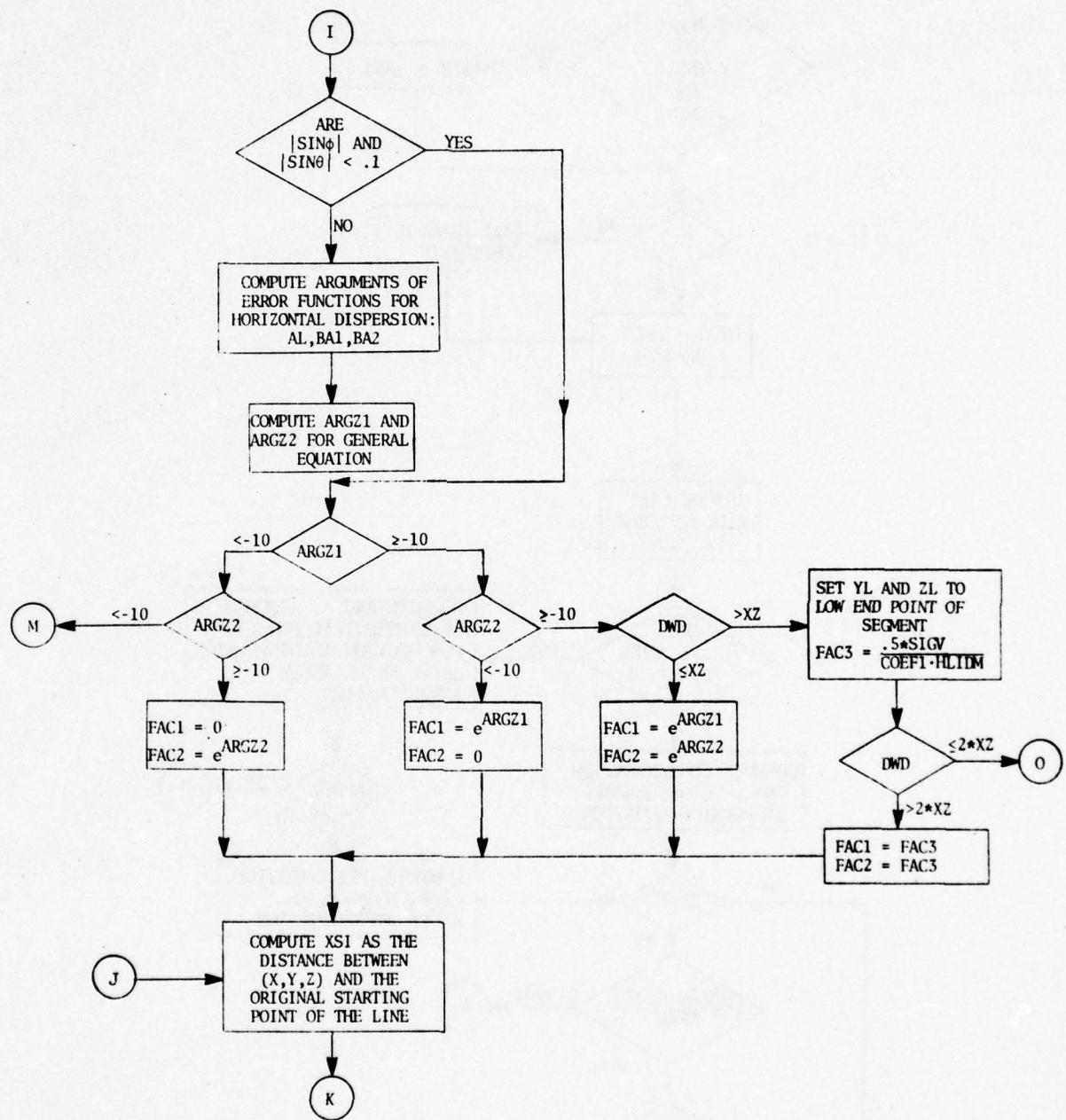


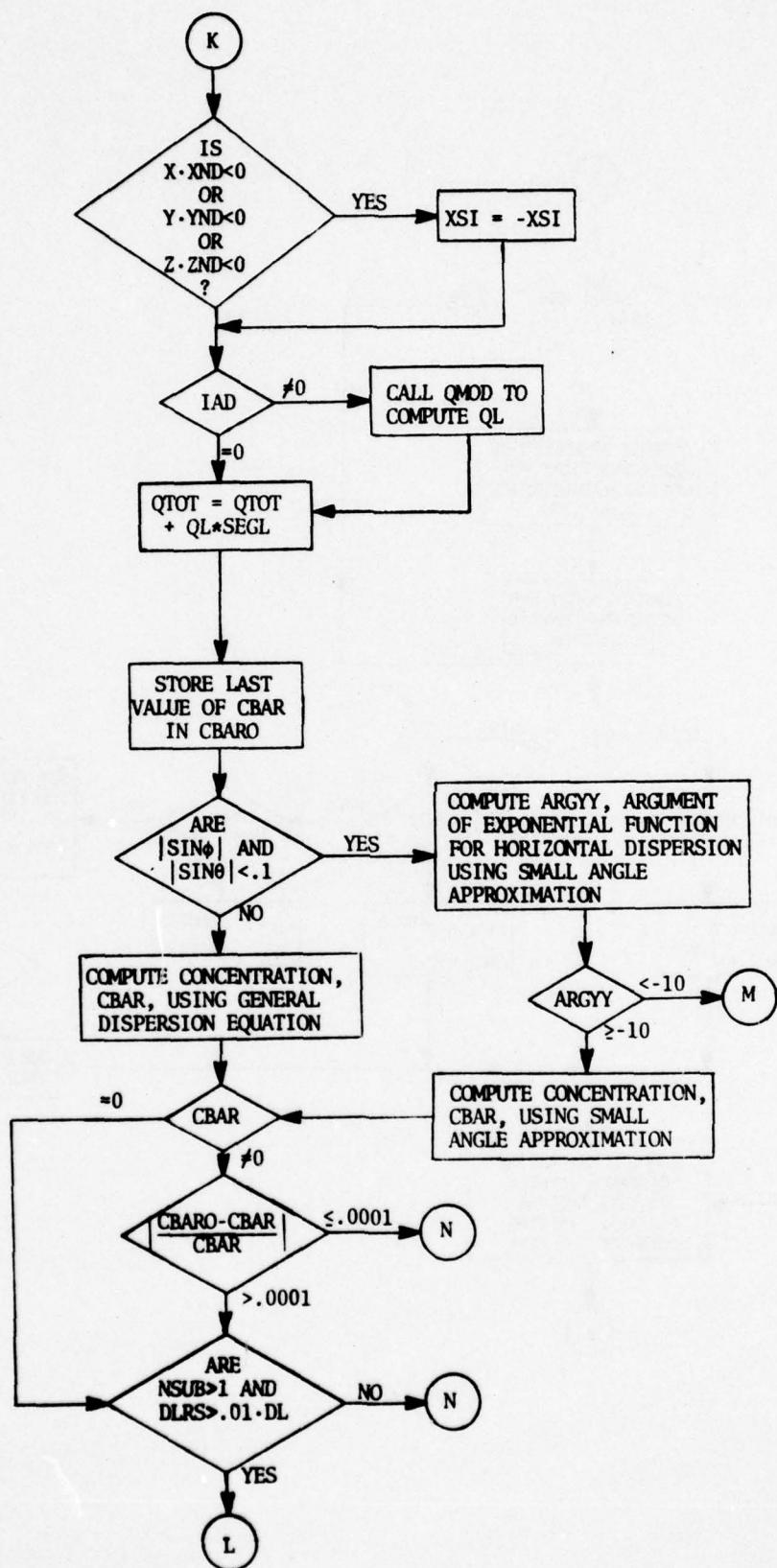


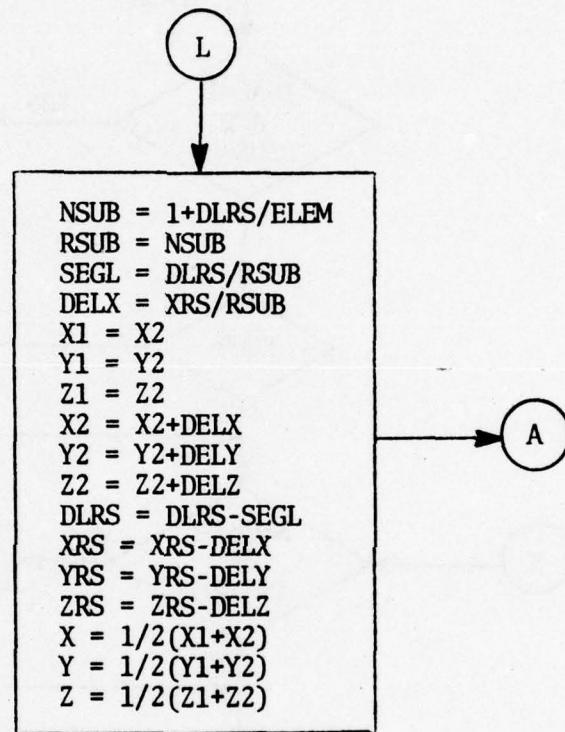


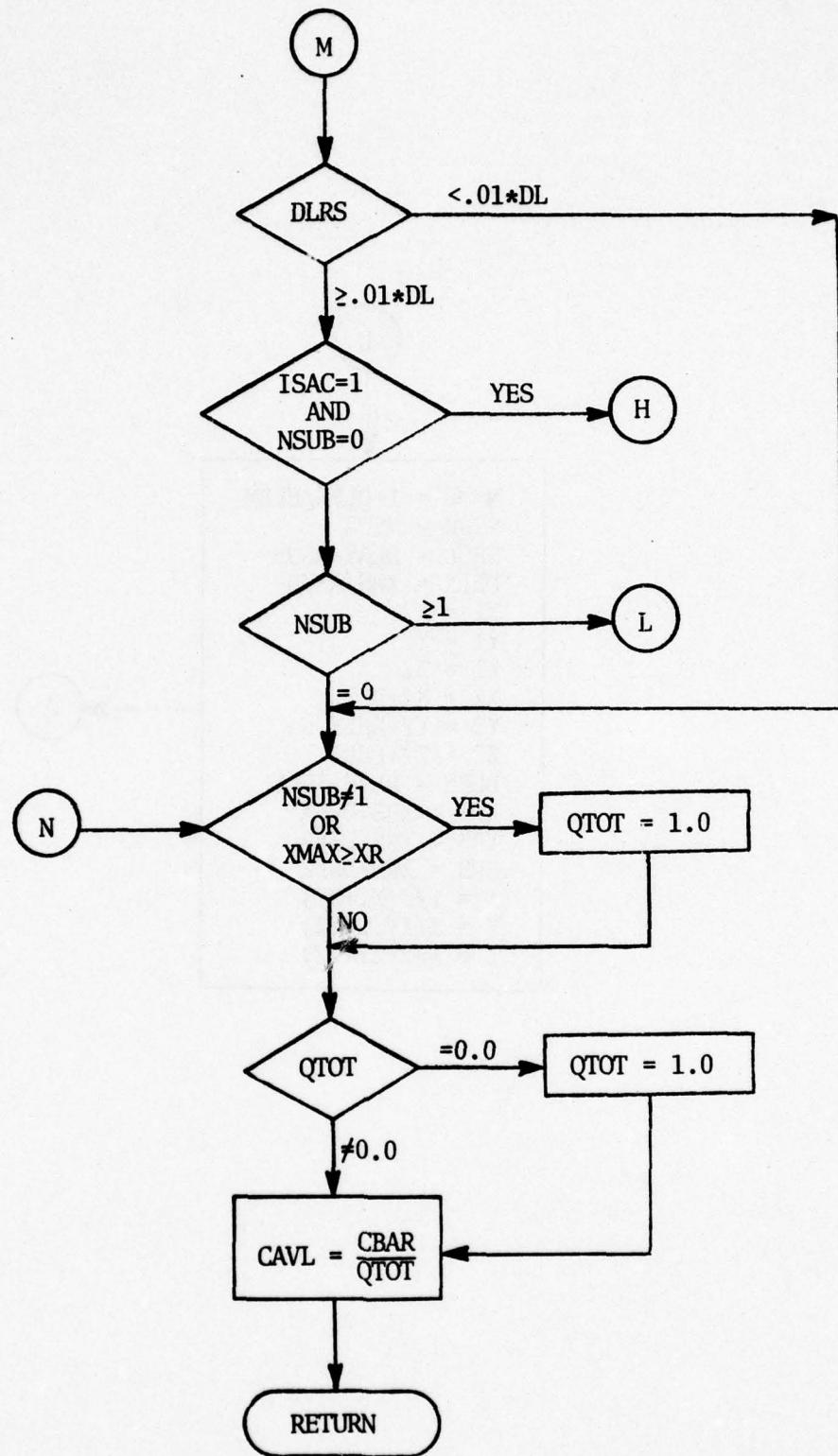


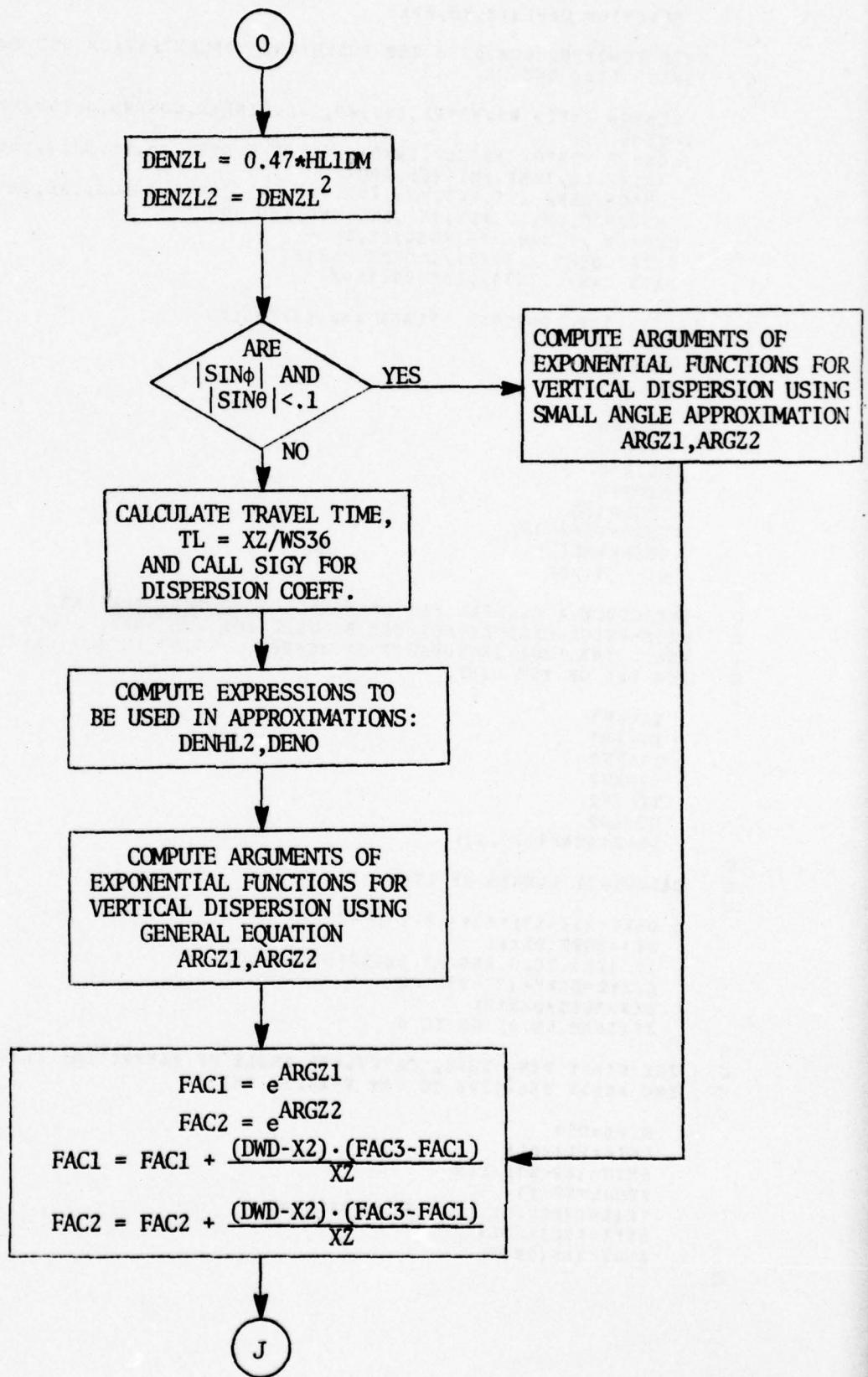












```

FUNCTION CAVL(XR,YR,ZR)                                CAVL0000
C THIS FUNCTION COMPUTES THE POLLUTANT CONCENTRATION DUE TO A CAVL0001
C FINITE LINE SOURCE CAVL0002
C
COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLTD,TEMF, CAVL0003
. TEMK
COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X5,Y5,Z5,W,DELZ,X6,Y6,Z6, CAVL0004
. V1,V2,DL,TIME,EMIS(6),NPOL CAVL0005
COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDCY,SUDOZ,IAD,TAIL,B,V12,VS, CAVL0006
. WS2,WSC,FR,SP,XST,YST,ZST,XND,YND,ZND CAVL0007
COMMON /XTRAN/ XZ,WSMD,TY,TZ CAVL0008
DATA COEF1 /.39894/,COEF2 /.31831/ CAVL0009
DATA CAN/0.7071/,EMIN/9.144/ CAVL0010
C
C INITIALIZE COUNTERS, FLAGS AND VARIABLES CAVL0011
C
ISUB=0                                              CAVL0012
NSUB=0                                              CAVL0013
ISAC=0                                              CAVL0014
LSAC=0                                              CAVL0015
CEAR=0.                                             CAVL0016
DWDA=0.                                             CAVL0017
QTOT=0.                                             CAVL0018
SEG1=1.0                                            CAVL0019
WS36=WS*3600.                                         CAVL0020
HLIDM=HLID                                           CAVL0021
QL = 1./DL                                           CAVL0022
C
C INTRODUCE A GENERAL SET OF NOTATION SO THAT THE SAME CAVL0023
C DISPERSION CALCULATION CAN BE USED FOR THE SMALL ANGLE CASE CAVL0024
C WHERE THE LINE IS FURTHER SEGMENTED. X1,Y1,Z1 NOW REFER TO THE CAVL0025
C LOW END OF THE LINE. CAVL0026
C
X1=XW1                                              CAVL0027
Y1=YW1                                              CAVL0028
Z1=ZW1                                              CAVL0029
X2=XW2                                              CAVL0030
Y2=YW2                                              CAVL0031
Z2=ZW2                                              CAVL0032
XMAX=AMAX1(X1,X2)                                    CAVL0033
C
C CALCULATE LENGTH OF LINE CAVL0034
C
5 DLXY=(X2-X1)**2+(Y2-Y1)**2 CAVL0035
DL1=SQRT(DLXY)                                       CAVL0036
IF (DL1.EQ.0.AND.Z1.EQ.Z2) GO TO 600 CAVL0037
DLXYZ=DLXY+(Z2-Z1)**2 CAVL0038
DLN=SQRT(DLXYZ)                                     CAVL0039
IF (ISUB.NE.0) GO TO 6 CAVL0040
C
C THE FIRST TIME THRU, CALCULATE ANGLE OF ELEVATION, THETA, CAVL0041
C AND ANGLE RELATIVE TO THE X-AXIS, PHI CAVL0042
C
DIRS=DL1                                             CAVL0043
CSTH=DL1/DLN                                         CAVL0044
SNTH=(Z2-Z1)/DLN                                     CAVL0045
PROJL=Y2-Y1                                         CAVL0046
IF (ABS(PROJL).LT.1.E-20) PROJL=0.                  CAVL0047
SNFI=PROJL/DL1                                       CAVL0048
ASNF=ABS(SNFI)                                      CAVL0049
C

```

```

C FIND HIGH AND LOW ENDS OF LINE AS PROJECTED ON THE X-Y PLANE
C
6 CONTINUE
  IF(Y1.GT.Y2) GO TO 1
  YH=Y2
  YL=Y1
  GC TO 2
1 YH=Y1
  YL=Y2
2 CONTINUE

C TEST THE RECEPTOR LOCATION RELATIVE TO THE LINE SOURCE
C
XMIN=AMIN1(X1,X2)
  IF ((XMIN-XR).GE.0.5) GO TO 500

C RECEPTOR IS DOWNWIND, FIND DISTANCE TO UPWIND END OF LINE
C
XFAR=XR-XMIN

C STOP PREVIOUS VALUES AND COMPUTE NEW DOWNWIND DISTANCES
C AND TRAVEL TIMES FOR PSEUDO UPWIND END OF LINE
C
DWDA0=DWDA
SIGFO=SIGF
DWDA=XFAR+SUD0Y
DWDB=XFAR+SUD0Z
TFAR=DWDA/WS36
TFBR=DWDB/WS36

C COMPUTE UPWIND END DISPERSION COEFFICIENTS
C
SIGF=SIGY(JSTAR,TFAR)
SIGFZ=SIGZ(JSTAR,TFBR)

C STOP LINE COORDINATES
C
IF(X1.LE.X2) GO TO 21
XA=X2
YA=Y2
XB=X1
YB=Y1
ZB=Z1
GC TO 22
21 XA=X1
YA=Y1
XB=X2
YB=Y2
ZP=Z2
22 CONTINUE
  IF(ISAC.EQ.1) GO TO 4

C ARE Y AND Z COORDS OF RECEPTOR WITHIN 4 TIMES THE DISPERSION
C COEFFICIENT OF THE Y AND Z COORDS OF THE LINE
C
IF(YF.GT.(YH+4.*SIGF)) GO TO 500
IF(YF.LT.(YL-4.*SIGF)) GO TO 500
IF(ZR.GT.(Z2+4.*SIGFZ)) GO TO 500
IF(ZR.LT.(Z1-4.*SIGFZ)) GO TO 500
IF (ASN.FLT. CAN .AND. ABS(SNTH) .LT. CAN) GO TO 3
IF (IAC.NE.0) GO TO 3

```

```

C ANGLE IS LARGE: ARE THE RECEPTOR COORDS WITHIN 3 TIMES THE CAVL0124
C DISPERSION COEFFICIENT OF THE LINE COORDS. CAVL0125
C
C IF(YR.GT.(YH+3.*SIGF)) GO TO 500 CAVL0126
C IF(YR.LT.(YL-3.*SIGF)) GO TO 500 CAVL0127
C IF(ZR.GT.(Z2+3.*SIGFZ)) GC TO 500 CAVL0128
C IF(ZR.LT.(Z1-3.*SIGFZ)) GC TO 500 CAVL0129
C
C SET (X,Y) AS POINT ON LINE WHERE IMPACT IS GREATEST CAVL0130
C
C X=X1+ (YR-Y1)*(X2-X1)/(Y2-Y1) CAVL0131
C IF(X.GT.XB) GC TO 333 CAVL0132
C IF(X.LT.XA) GC TO 33 CAVL0133
C Y=YF CAVL0134
C GO TO 4 CAVL0135
C
C ANGLE IS SMALL: REDEFINE LINE COORDS AND SET (X,Y) AS CAVL0136
C MIDPOINT OF SEGMENT CAVL0137
C
C 3 IF(ASN.F.LT.0.1.AND.(ABS(SNTH)).LT.0.1) LSAC=1 CAVL0138
C ISAC=1 CAVL0139
30 X=A MIN1(XP,XA) CAVL0140
Y1=Y1+ (X-X1)*(Y2-Y1)/(X2-X1) CAVL0141
Z1=Z1+ (Z-Z1)*(Z2-Z1)/(X2-X1) CAVL0142
X1=X CAVL0143
X2=XB CAVL0144
Y2=YB CAVL0145
Z2=ZB CAVL0146
X=0.5*(X1+X2) CAVL0147
Y=0.5*(Y1+Y2) CAVL0148
GC TO 5 CAVL0149
33 X=XA CAVL0150
Y=YA CAVL0151
GC TO 4 CAVL0152
333 X=XB CAVL0153
Y=YB CAVL0154
C
C COMPUTE DOWNWIND DISTANCE CAVL0155
C
C 4 DWD=XR-X CAVL0156
IF (DWD.LT.-.01) GO TO 30 CAVL0157
DWD1=DWD CAVL0158
IF (ISAC.NE.1) GO TO 40 CAVL0159
DWD1=XP-X1 CAVL0160
IF (NSUB.LE.1) DWD=DWD1 CAVL0161
C
C COMPUTE ESEUDO DOWNWIND DISTANCES CAVL0162
C
C 40 DWDY1=DWD1+SUD0Y CAVL0163
DWDY=DWD+SUD0Y CAVL0164
DWDZ=DWD+SUD0Z CAVL0165
C
C SET Z COORDINATE OF LINE CAVL0166
C
IF(X1.EQ.X2) GO TO 44 CAVL0167
Z=Z1+ (X-X1)*(Z2-Z1)/(X2-X1) CAVL0168
GC TO 444 CAVL0169
44 Z=Z1+ (Y-Y1)*(Z2-Z1)/(Y2-Y1) CAVL0170
444 CONTINUE CAVL0171
C
C COMPUTE TRAVEL TIME AND DISPERSION COEFFICIENT FOR CAVL0172
C PSEUDO DOWNWIND DISTANCE CAVL0173
C CAVL0174
C CAVL0175
C CAVL0176
C CAVL0177
C CAVL0178
C CAVL0179
C CAVL0180
C CAVL0181
C CAVL0182
C CAVL0183
C CAVL0184
C CAVL0185

```

```

IF (DWDY1.EQ.DWDA) GO TO 4111
IF (DWDY1.EQ.DWDAD) GO TO 4113
THFH1=DWDY1/WS36
SIGH1=SIGY(JSTAB,THFH1)
4211 CONTINUE
C
C DETERMINE FACTOR TO BE USED IN SUB-DIVIDING THE LINE
C
ELEM=AMAX1(0.2*DWDY1,SIGH1)
IF (IAD.NE.0) ISAC=1
IF (ISAC.EQ.1) ELEM=.1*ELEM
IF (ELEM.LT.EMIN) ELEM=EMIN
C
C BRANCH IF ANGLE IS SMALL AND LINE SOURCE IS LONG.
C
IF (IAD.NE.0) GO TO 4311
IF (ASN.FE.CAN) GO TO 4312
4311 IF (DL1.GT.(1.5*ELEM)) GO TO 55
C
C COMPUTE TRAVEL TIME AND DISPERSION COEFFICIENT FOR PSEUDO
C VERTICAL DISTANCE
C
4312 IF (DWDY.EQ.DWDY1) GO TO 4112
THFH=DWDY/WS36
SIGH=SIGY(JSTAB,THFH)
4212 CCNTINUE
THPV=DWDZ/WS36
SIGV=SIGZ(JSTAB,THPV)
C
C EXPRESSIONS TO BE USED IN APPROXIMATIONS
C
DENH2=2.*SIGH**2
DENZ2=2.*SIGV**2
D=SIGH*SIGV
C
C ARGUMENTS OF EXPONENTIAL FUNCTION FOR VERTICAL DISPERSION
C USING SMALL ANGLE APPROXIMATION
C
ARGZ1=-(ZR-Z1)**2/DENZ2
ARGZ2=-(ZR+Z1)**2/DENZ2
IF (LSAC.EQ.1) GO TO 446
GC TO 445
4111 SIGH1=SIGF
GO TO 4211
4112 SIGH=SIGH1
GO TO 4212
4113 SIGH1=SIGFO
GO TO 4211
C
C LARGE ANGLE CASE: ARGUMENTS OF ERROR FUNCTIONS FOR
C HORIZONTAL DISPERSION
C
445 CCNTINUE
APG=CSTH**2*SNFI**2*SIGV**2+SNTH**2*SIGH**2
RARG=SQRT(ARG)
A=FAFG/(1.4142*D)
AL=DLN*A
ARG1=(YR-Y1)*CSTH*SNFI*SIGV**2
ARG21=(ZR-Z1)*SNTH*SIGH**2
ARG22=-(ZR+Z1)*SNTH*SIGH**2
PA1=-(ARG1+ARG21)/(1.4142*D*RAPG)

```

```

BA2=-(ARG1+ARG22)/(1.4142*D*BAR)
C1=(YR-Y1)**2/DENH2-ARGZ1
C2=(YR-Y1)**2/DENH2-ARGZ2
CAVI0248
CAVI0249
CAVI0250
CAVI0251
CAVI0252
CAVI0253
CAVI0254
CAVI0255
CAVI0256
CAVI0257
CAVI0258
CAVI0259
CAVI0260
CAVI0261
CAVI0262
CAVI0263
CAVI0264
CAVI0265
CAVI0266
CAVI0267
CAVI0268
CAVI0269
CAVI0270
CAVI0271
CAVI0272
CAVI0273
CAVI0274
CAVI0275
CAVI0276
CAVI0277
CAVI0278
CAVI0279
CAVI0280
CAVI0281
CAVI0282
CAVI0283
CAVI0284
CAVI0285
CAVI0286
CAVI0287
CAVI0288
CAVI0289
CAVI0290
CAVI0291
CAVI0292
CAVI0293
CAVI0294
CAVI0295
CAVI0296
CAVI0297
CAVI0298
CAVI0299
CAVI0300
CAVI0301
CAVI0302
CAVI0303
CAVI0304
CAVI0305
CAVI0306
CAVI0307
CAVI0308
CAVI0309

C ARGUMENTS OF EXPONENTIAL FUNCTIONS FOR VERTICAL DISPERSION
C USING THE GENERAL EQUATION
C
C ARGZ1=BA1**2-C1
C ARGZ2=BA2**2-C2
446 IF(ARGZ1.LT.-10.) GOTO 2411
IF(ARGZ2.GE.-10.) GOTO 2412
FAC1=EXP(ARGZ1)
FAC2=0
GC TO 39
2411 IF (ARGZ2.LT.-10) GO TO 500
FAC1=0
FAC2=EXP(ARGZ2)
GC TO 39
2412 IF(DWD.GT.XZ) GO TO 100
C
C DOWNWIND DISTANCE IS LESS THAN THE CRITICAL DISTANCE: ONLY
C SOURCE AND GROUND REFLECTION ARE CONSIDERED
C
FAC1=EXP(ARGZ1)
FAC2=EXP(ARGZ2)
39 CCNTINUE
C
C FIND THE LINEAR DISTRIBUTION OF POLLUTION ON THE
C RUNWAY FOR LANDING AND TAKE-OFF
C
XSI2=(X-XST)**2+(Y-YST)**2+(Z-ZST)**2
XSI=SQRT(XSI2)
IF(X*XND.LT.0.OF.Y*YND.LT.0.OF.Z*ZND.LT.0) XSI=-XSI
IF (IAD.NE.0) CALL QMOD(XSI,QL)
QTOT=QTOT+QL*SEGL
C
C STORE LAST VALUE OF CBAR
C
CBAR=CBAR
IF(LSAC.EQ.1) GO TO 50
C
C GENERAL DISPERSION EQUATION
C
FJ1=FAC1*DIFEFF(BA1,AL)
FJ2=FAC2*DIFEFF(BA2,AL)
CBAR=CBAR+0.35355*COEFF1*QI*(FJ1+FJ2)/(A*D)
499 IF (CBAR0.EQ.0) GO TO 49
IF(AES((CBAR0-CBAR)/CBAR).LE..00010) GO TO 600
49 CCNTINUE
IF(NSUB.GT.1.AND.DLRS.GT.(.01*DL)) GO TO 60
GC TO 600
C
C SMALL-ANGLE APPROXIMATION
C
50 ARGYY=-(YR-Y1)**2/DENH2
IF(ARGYY.LT.-10.) GO TO 500
FAC=0.5*(FAC1+FAC2)
BRAC=EXP(ARGYY)
CEAF=CBAR+COEFF2*QL*DLN*FAC*BRAC/D
GO TO 499
C
C ANGLE IS SMALL AND SOURCE IS LONG

```

```

C
55  ISUR=1
    XRS=X2-X1
    YPS=Y2-Y1
    ZRS=Z2-Z1
    X2=X1
    Y2=Y1
    Z2=Z1
    C
    C COMPUTE COORDINATES FOR NEXT LINE SEGMENT
    C
60  NSUB=1.+DLRS/FLEM
    RSUB=NSUB
    SEGI=DLRS/RSUB
    DELX=XPS/FSUB
    DELY=YRS/FSUB
    DELZ=ZRS/RSUB
    X1=X2
    Y1=Y2
    Z1=Z2
    X2=X2+DELX
    Y2=Y2+DELY
    Z2=Z2+DELZ
    DLPS=DLRS-SEGL
    XFS=XFS-DELX
    YRS=YRS-DELY
    ZRS=ZRS-DELZ
    X=.5*(X1+X2)
    Y=.5*(Y1+Y2)
    Z=.5*(Z1+Z2)
    C
    C GO BACK TO COMPUTE CONTRIBUTION FROM NEXT SEGMENT
    C
    GC TC 5
    C
    C DOWNWIND DISTANCE IS GREATER THAN, BUT LESS THAN TWICE, THE
    C CRITICAL DISTANCE. LINEAR INTERPOLATION IS USED
    C
100  YL = Y1
    ZL = Z1
    IF (Z1 .LE. Z2) GO TO 105
    YL = Y2
    ZL = Z2
105  FAC3=0.5*SIGV/(COEF1*HLIDM)
    IF (LWD.GT.2.*XZ) GO TO 200
    DENZI=0.47*HLIDM
    DENZI2=DENZL**2
    IF (ISAC.EQ.1) GO TO 101
102  TI=XZ/WS36
    DENH12=2.*STGY(JSTAB,TL)**2
    DENO=CSTH**2*SNFI**2*DENZI2 + SNTH**2*DENHL2
    AFGZ1=-((YR-YL)*SNTH-(ZR-ZL)*CSTH*SNFI)**2/DENO
    AFGZ2=-((YR-YL)*SNTH-(ZR+ZL)*CSTH*SNFI)**2/DENO
    GC TC 103
101  AFGZ1=- (ZR-ZL)**2/DENZL2
    AFGZ2=- (ZR+ZL)**2/DENZL2
103  FAC1=EXP(AGZ1)
    FAC2=EXP(AGZ2)
    FAC1=FAC1+(DWD-XZ)*(FAC3-FAC1)/XZ
    FAC2=FAC2+(DWD-XZ)*(FAC3-FAC2)/XZ
    GC TC 39
    C

```

```

C      DOWNWIND DISTANCE IS BEYOND 2 TIMES THE CRITICAL DISTANCE,      CAVL0372
C      UNIFORM MIXING IS ASSUMED          CAVL0373
C
C      200 FAC1=FAC3          CAVL0374
      FAC2=FAC3          CAVL0375
      GO TO 39          CAVL0376
      500 IF(DLRS.LT. (.01*DL)) GO TO 600          CAVL0377
      IF(ISAC.EQ.1.AND.NSUB.EQ.0) GO TO 55          CAVL0378
      IF(NSUB.GE.1) GO TO 60          CAVL0379
      600 IF (NSUB.NE.1.OR.XMAX.GE.XF) QTOT=1.0          CAVL0380
      IF (QTOT.EQ.0.0) QTOT=1.0          CAVL0381
C
C      NORMALIZE CBAR TO THE TOTAL POLLUTANT DENSITY CALCULATED      CAVL0382
C      ALONG THE LINE          CAVL0383
C
C      CAVL=CBAR/QTOT          CAVL0384
      RETURN          CAVL0385
      END          CAVL0386
      CAVL=CBAR/QTOT          CAVL0387
      RETURN          CAVL0388
      END          CAVL0389

```

## SUBROUTINE CLASSE

### Purpose:

To print an error message if the wrong ICLASS value is input to one of the airbase non-aircraft or environ emission distribution subroutines.

### Input:

None

### Output:

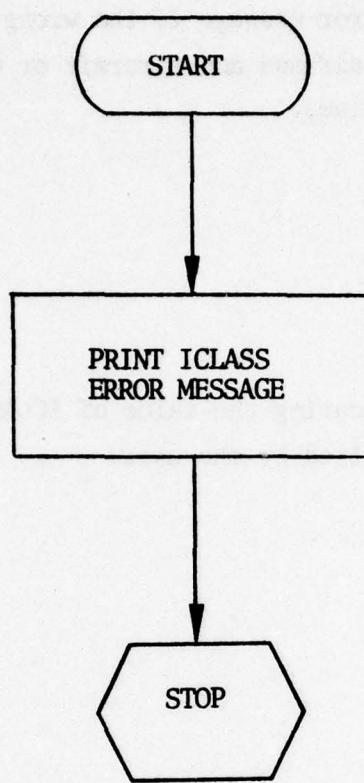
A message indicating the value of ICLASS set by the code and the value supplied by the user.

### Subroutines

#### Called:

None

SUBROUTINE CLASSE



C SUBFCUTINE CLASSE (I,J)  
C THIS FCUTINE PRINTS THE ICLASS ERROR MESSAGE  
C  
PRINT 1, I,J  
1 FORMAT(17HOICLASS SHOULD BE,I4,18H, INPUT CARD READS,I4)  
STOP  
END

CLASE000  
CLASE001  
CLASE002  
CLASE003  
CLASE004  
CLASE005  
CLASE006  
CLASE007

## SUBROUTINE DEPART

### Purpose:

To calculate the points in the runway roll and climbout modes as a function of aircraft type using current meteorological conditions and airbase specific pressure altitudes and airbase dependent basic aircraft parameters.

### Input:

Basic aircraft data, current meteorological conditions, runway data, aircraft identification.

### Output:

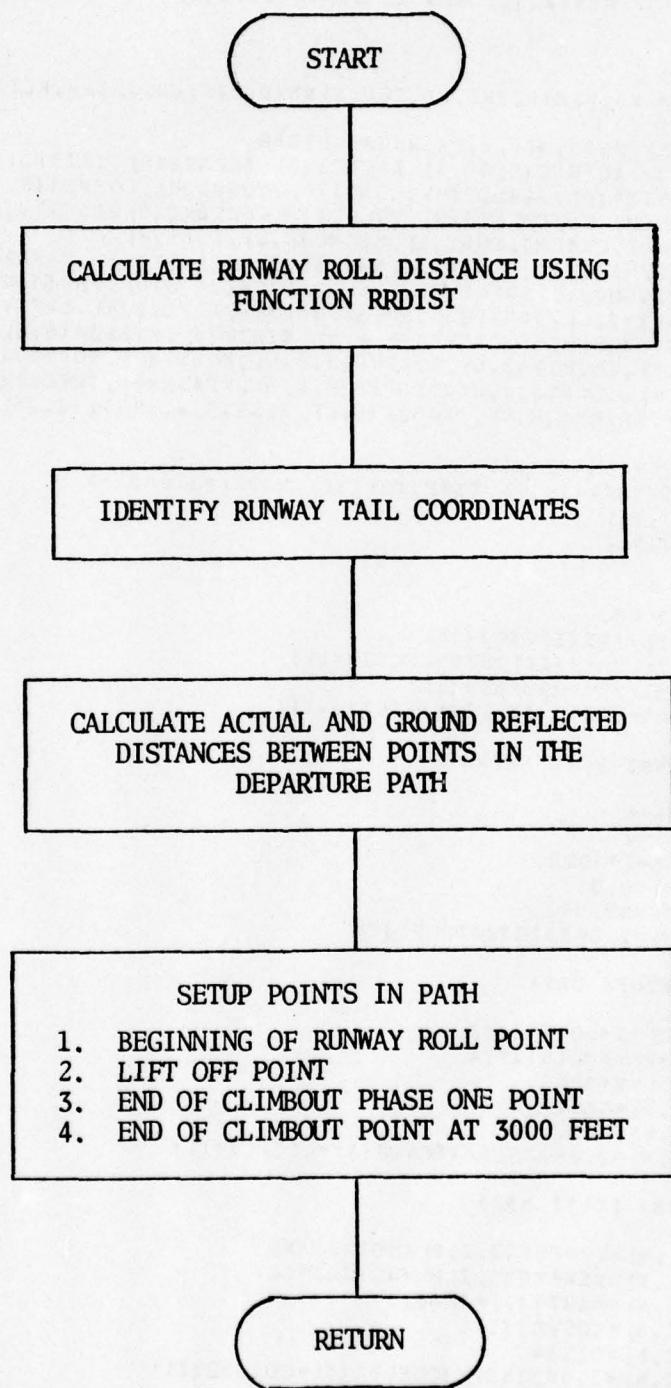
Points in departure path as a function of runway and aircraft type.

### Subroutine

### Called:

Function RRDIST.

SUBROUTINE DEPART



```

C SUBROUTINE DEPART(N,I) DEPRT000
C THIS ROUTINE CALCULATES THE POINTS IN THE DEPARTURE PATH DEPRT001
C AS A FUNCTION OF RUNWAY(N) AND AIRCRAFT TYPE(I) DEPRT002
C
C REAL INDSPD DEPRT003
C INTEGER ENGNO DEPRT004
C COMMON / MET / WS,WSMEH,IWS,WD,IWD,SINNEWD,CVSEWD,JSTAB,HLID,TEMF, DEPRT005
C . TEMK DEPRT006
C CMMCN /ANMNET/ TBAR,ACD,P,PA,WSBAR,DTBAR DEPRT007
C COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8), DEPRT008
C . INDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8), DEPRT009
C . COSPD2(8),SRTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8), DEPRT011
C . APEHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2),IDRR(8), DEPRT012
C COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNAFR(8), DEPRT013
C . ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8), DEPRT014
C . DISFNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8), DEPRT015
C . ACSPIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6), DEPRT016
C . TIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBT(6),NOBSEG(8,6), DEPRT017
C . IOBSEG(16,8,6),IDOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6), DEPRT018
C . PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JFS1(8) DEPRT019
C RD=RNWY(7,N) DEPRT020
C WSPD=WS*1.9426*COS(WD-RD) DEPRT021
C HDIS12=RRDIST(IDRR(I),PA,TEMF,TOWT(I),WSPD)*3.048E-4 DEPRT023
C XA=SIN(RNWY(7,N)) DEPRT024
C YA=COS(RNWY(7,N)) DEPRT025
C X=RNWY(2,N) DEPRT026
C Y=RNWY(3,N) DEPRT027
C Z=RNWY(4,N)/1000. DEPRT028
C DIS23=COHT1(I)/SIN(ASCNT1(I)) DEPRT029
C DIS34=(CLMBHT-COHT1(I))/SIN(ASCNT2(I)) DEPRT030
C HDIS23=COHT1(I)/TAN(ASCNT1(I)) DEPRT031
C HDIS34=(CLMBHT-COHT1(I))/TAN(ASCNT2(I)) DEPRT032
C
C START OF RUNWAY ROLL DATA DEPRT033
C
C DEPFCN(1,I,N)=X DEPRT034
C DEPFCN(2,I,N)=Y DEPRT035
C DEPFCN(3,I,N)=Z*1000. DEPRT036
C DEPFCN(4,I,N)=0.0 DEPRT037
C DEPFCN(5,I,N)=HDIS12 DEPRT038
C DEPFCN(6,I,N)=2.0*HDIS12/TOSPD(I) DEPRT039
C
C ECINT OF LIFTOFF DATA DEPRT040
C
C DEPFCN(7,I,N)=X+HDIS12*XA DEPRT041
C DEPFCN(8,I,N)=Y+HDIS12*YA DEPRT042
C DEPFCN(9,I,N)=Z*1000. DEPRT043
C DEPFCN(10,I,N)=TOSPD(I) DEPRT044
C DEPFCN(11,I,N)=DIS23 DEPRT045
C DEPFCN(12,I,N)=2.0*DIS23/(TOSPD(I)+COSPD1(I)) DEPRT046
C
C END OF CLIMB1 ECINT DATA DEPRT047
C
C DEPFCN(13,I,N)=DEPFCN(7,I,N)+HDIS23*XA DEPRT048
C DEPFCN(14,I,N)=DEPFCN(8,I,N)+HDIS23*YA DEPRT049
C DEPFCN(15,I,N)=COHT1(I)*1000. DEPRT050
C DEPFCN(16,I,N)=COSPD1(I) DEPRT051
C DEPFCN(17,I,N)=DIS34 DEPRT052
C DEPFCN(18,I,N)=2.0*DIS34/(COSPD1(I)+COSPD2(I)) DEPRT053
C
C END OF CLIMBOUT POINT DATA DEPRT054
C

```

C

DEPFCN(19,I,N)=DEPFCN(13,I,N)+HDIS34\*XA  
DEPFCN(20,I,N)=DEPFCN(14,I,N)+HDIS34\*YA  
DEPFCN(21,I,N)=CLMBHT\*1000.  
DEPFCN(22,I,N)=COSPD2(I)  
RETUEN  
END

DEPRT062  
DEPRT063  
DEPRT064  
DEPRT065  
DEPRT066  
DEPRT067  
DEPRT068

FUNCTION DIFERF(X,PH)

Purpose:

To find the difference between two error functions,  $\text{erf}(X+PH) - \text{erf}(X)$ .

Input:

X and PH

Output:

The difference between the error functions

Procedure:

1. If  $PH \leq .05$ , the formula given in the Handbook of Mathematical Functions, National Bureau of Standards, Applied Mathematics Series 55 is used:

$$\text{DIFERF} = 1.12838 \cdot PH \cdot e^{-X^2} [1 - PH \cdot X + (2 \cdot X^2 - 1) \cdot PH^{2/3}]$$

2. If  $PH > .05$  and X and  $X+PH$  are of different sign:

$$\text{DIFERF} = \text{erf}(X+PH) - \text{erf}(X)$$

3. If  $PH > .05$  and X and  $X+PH$  are both negative:

$$\text{DIFERF} = -1 \cdot [\text{erfc}(-X) - \text{erfc}(-X-PH)]$$

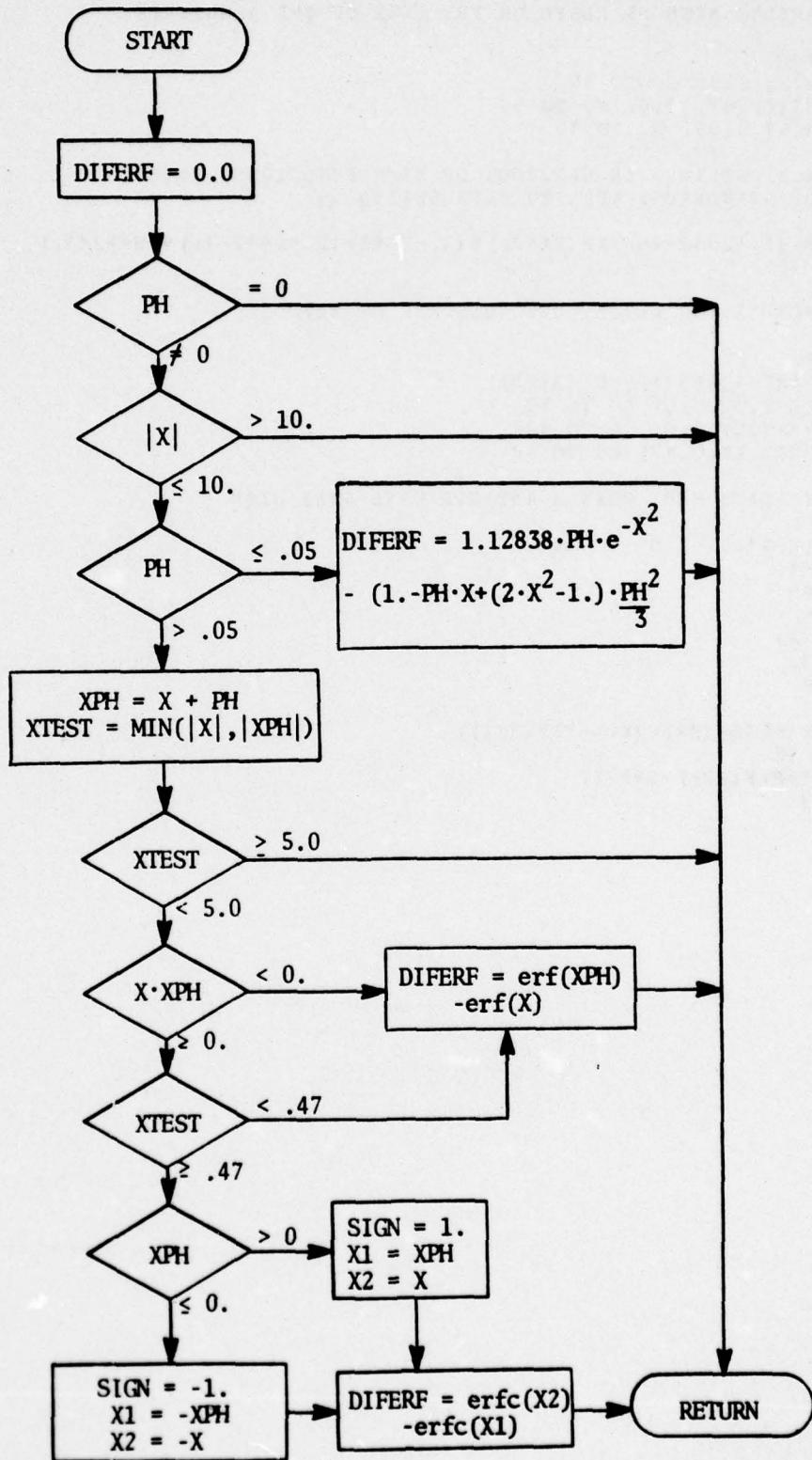
4. If  $PH > .05$  and X and  $X+PH$  are both positive:

$$\text{DIFERF} = \text{erfc}(X) - \text{erfc}(X+PH)$$

Function  
Called:

ERF, ERFC

FUNCTION DIFERF(X,PH)



```

FUNCTION DIFERF(X,PH) DIFER000
C DIFER001
C THIS FUNCTION FINDS THE DIFFERENCE BETWEEN TWO ERROR FUNCTIONS DIFER002
C USING VARYING METHODS BASED ON THE SIZE OF THE ARGUMENTS DIFER003
C DIFER004
C DIFER005
C DIFER006
C DIFER007
C DIFER008
C DIFER009
C DIFER010
C DIFER011
C DIFER012
C DIFER013
C DIFER014
C DIFER015
C DIFER016
C DIFER017
C DIFER018
C DIFER019
C DIFER020
C DIFER021
C DIFER022
C DIFER023
C DIFER024
C DIFER025
C DIFER026
C DIFER027
C DIFER028
C DIFER029
C DIFER030
C DIFER031
C DIFER032
C DIFER033
C DIFER034
C DIFER035
C DIFER036
C DIFER037
C DIFER038

DIFERF=0. DIFER000
IF (PH.EQ.0.0) GO TO 50 DIFER001
IF (ABS(X).GT.10.0) GO TO 50 DIFER002
IF (PH.GT.0.05) GO TO 10 DIFER003
DIFER004
DIFER005
DIFER006
DIFER007
DIFER008
DIFER009
DIFER010
DIFER011
DIFER012
DIFER013
DIFER014
DIFER015
DIFER016
DIFER017
DIFER018
DIFER019
DIFER020
DIFER021
DIFER022
DIFER023
DIFER024
DIFER025
DIFER026
DIFER027
DIFER028
DIFER029
DIFER030
DIFER031
DIFER032
DIFER033
DIFER034
DIFER035
DIFER036
DIFER037
DIFER038

USE METHOD OUTLINED IN HANDBOOK OF MATH FUNCTIONS, NATL DIFER010
BUREAU OF STANDARDS, APPLIED MATH SERIES 55 DIFER011
DIFER012
DIFER013
DIFER014
DIFER015
DIFER016
DIFER017
DIFER018
DIFER019
DIFER020
DIFER021
DIFER022
DIFER023
DIFER024
DIFER025
DIFER026
DIFER027
DIFER028
DIFER029
DIFER030
DIFER031
DIFER032
DIFER033
DIFER034
DIFER035
DIFER036
DIFER037
DIFER038

DIFERF=(1.12838*PH/EXP(X**2))*(1.-PH*X+(2.*X**2-1.)*PH**2/3.) GO TO 50 DIFER013
DIFER014
DIFER015
DIFER016
DIFER017
DIFER018
DIFER019
DIFER020
DIFER021
DIFER022
DIFER023
DIFER024
DIFER025
DIFER026
DIFER027
DIFER028
DIFER029
DIFER030
DIFER031
DIFER032
DIFER033
DIFER034
DIFER035
DIFER036
DIFER037
DIFER038

DIFFERENCE IS TOO LARGE, MUST USE ERF OR ERFC DIFER015
DIFER016
DIFER017
DIFER018
DIFER019
DIFER020
DIFER021
DIFER022
DIFER023
DIFER024
DIFER025
DIFER026
DIFER027
DIFER028
DIFER029
DIFER030
DIFER031
DIFER032
DIFER033
DIFER034
DIFER035
DIFER036
DIFER037
DIFER038

10 XPH=X+PH DIFER018
XTEST=AMIN1(ABS(X),ABS(X+PH)) DIFER019
IF (XTEST.GE.5.0) GO TO 50 DIFER020
IF (X*XPH.LT.0.0) GO TO 40 DIFER021
IF (XTEST.LT.0.47) GO TO 40 DIFER022
DIFER023
DIFER024
DIFER025
DIFER026
DIFER027
DIFER028
DIFER029
DIFER030
DIFER031
DIFER032
DIFER033
DIFER034
DIFER035
DIFER036
DIFER037
DIFER038

DIFER023
DIFER024
DIFER025
DIFER026
DIFER027
DIFER028
DIFER029
DIFER030
DIFER031
DIFER032
DIFER033
DIFER034
DIFER035
DIFER036
DIFER037
DIFER038

C CAN ONLY REACH HERE WHEN X AND XPH HAVE SAME SIGN DIFER024
C DIFER025
C DIFER026
C DIFER027
C DIFER028
C DIFER029
C DIFER030
C DIFER031
C DIFER032
C DIFER033
C DIFER034
C DIFER035
C DIFER036
C DIFER037
C DIFER038

IF (XPH.GT.0.0) GO TO 20 DIFER026
SIGN=-1. DIFER027
X1=-XPH DIFER028
X2=-X DIFER029
GO TO 30 DIFER030
DIFER031
DIFER032
DIFER033
DIFER034
DIFER035
DIFER036
DIFER037
DIFER038

20 SIGN=1. DIFER027
X1=XPH DIFER028
X2=X DIFER029
GO TO 30 DIFER030
DIFER031
DIFER032
DIFER033
DIFER034
DIFER035
DIFER036
DIFER037
DIFER038

30 DIFERF=SIGN*(ERFC(X2)-ERFC(X1)) DIFER031
GO TO 50 DIFER032
DIFER033
DIFER034
DIFER035
DIFER036
DIFER037
DIFER038

40 DIFERF=ERF(XPH)-ERF(X) DIFER031
50 RETURN DIFER032
END DIFER033
DIFER034
DIFER035
DIFER036
DIFER037
DIFER038

```

## SUBROUTINE EMISAR

### Purpose:

To add emissions from a given activity to all others contained in the specified geometric area or line.

### Input:

None

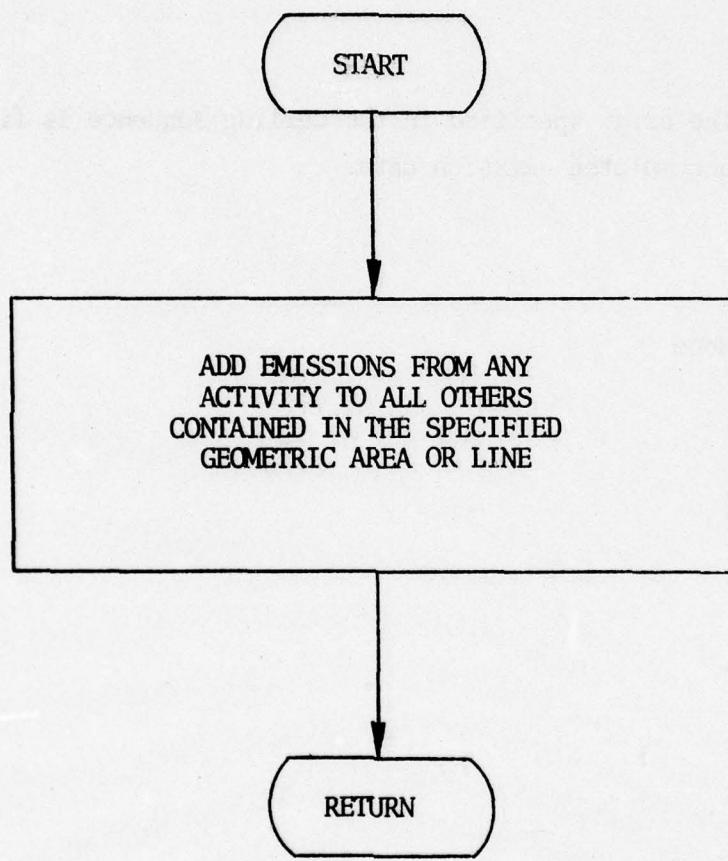
### Output:

The array specified in the calling sequence is filled with the accumulated emission data.

### Subroutines Called:

None

SUBROUTINE EMISAR



```

SUBROUTINE EMISAR(MAXN,ARRAY,I1,I2)          EMISR000
C
C   THIS ROUTINE ACCUMULATES EMISSIONS FROM ANY ACTIVITY WITH    EMISR001
C   OTHERS CONTAINED IN THE SAME AIRBASE AREA OR LINE.           EMISR002
C   MAXN      = NO. OF SOURCES IN AN ACTIVITY                   EMISR003
C   ARRAY     = SPECIFIED AREA OR LINE OUTPUT ARRAY             EMISR004
C   I1,I2     = DIMENSIONS OF ARRAY                           EMISR005
C   NSPCE     = POINTER TO LOCATION OF SOURCES IN SORCE       EMISR006
C   LCC1      = POINTER TO LOCATION OF LIST OF EMISSIONS IN ARRAY EMISR007
C   SORCE(2,N) = POINTER TO LOCATION OF SOURCE AREA OR LINE    EMISR008
C
C   COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,    EMISR009
C   . NACET,NACAR,NACLN,ENET(16,100),ENAR(11,100),ENLN(14,20),    EMISR010
C   . ABFT(16,150),ABAR(11,100),ABLN(14,100)                   EMISR011
C   COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)    EMISR012
C   . ,LCC1,LCC2,NGEOM,IPT                                     EMISR013
C   DIMENSION ARRAY(I1,I2)                                     EMISR014
C   LSRCE=NSRCE+1                                         EMISR015
C   NSRCE=NSRCE+MAXN                                     EMISR016
C
C   DC 10 N=LSRCE,NSRCE                                 EMISR017
C   J=SORCE(2,N)                                         EMISR018
C   DC 10 I=1,NPLTS                                     EMISR019
C   ARRAY(I+LOC1,J)=ARRAY(I+LCC1,J)+SORCE(I+2,N)          EMISR020
10  CCNTINUE                                         EMISR021
      RETURN                                         EMISR022
      END                                           EMISR023
                                                EMISR024
                                                EMISR025
                                                EMISR026

```

## SUBROUTINE ENARAY

### Purpose:

1. To read from the master source tape all data needed to define environ point, area and line sources.
2. To compute the emission rates due to point sources, stationary, mobile, land use or combined area sources and roadway and non-roadway line sources.

### Input:

If the diurnal distribution cards are input, an additional parameter, IMETH, is input here to choose the method of distribution of emissions from those land use or combined area source activities not using the default of a uniform distribution.

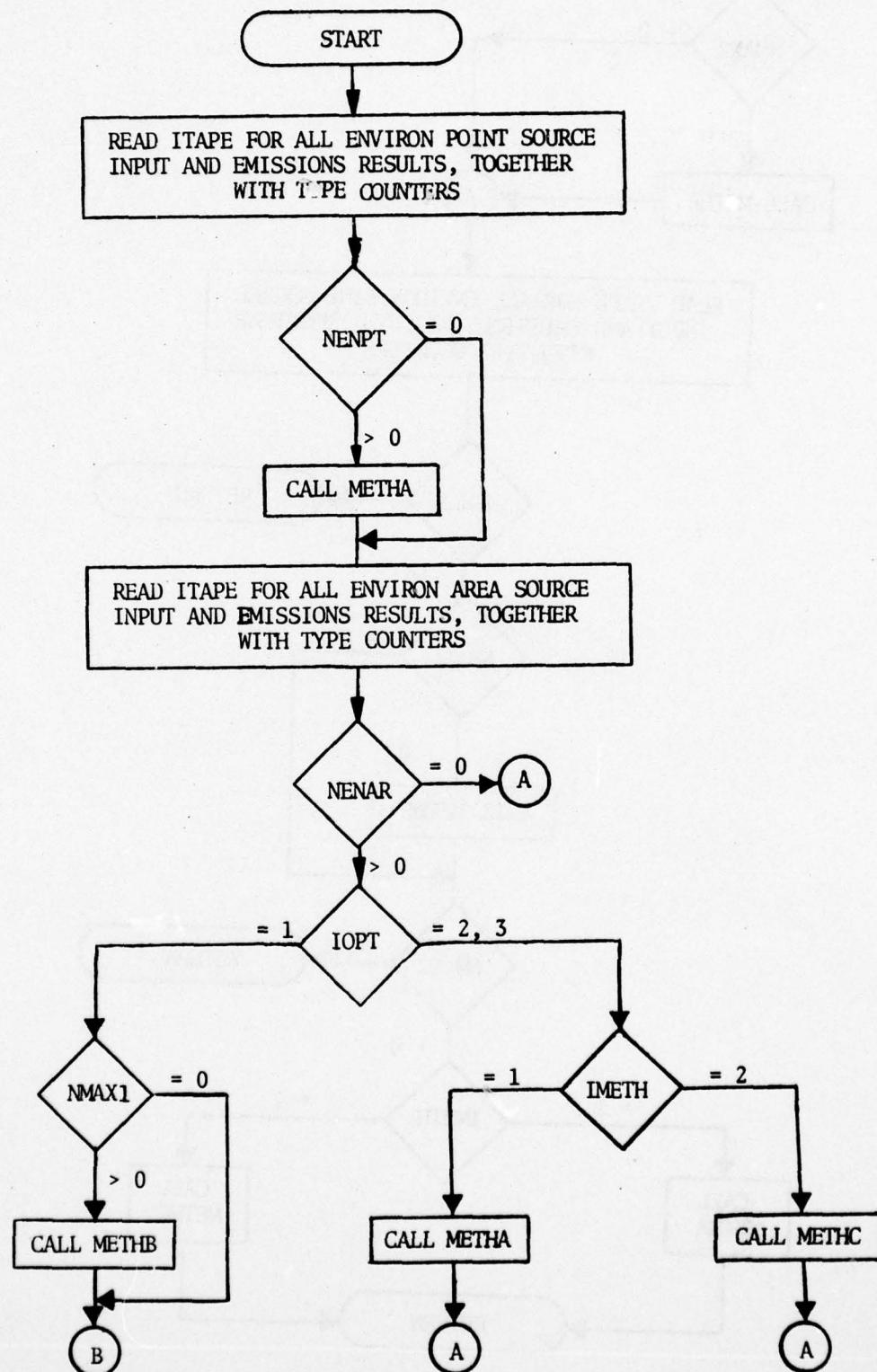
### Output:

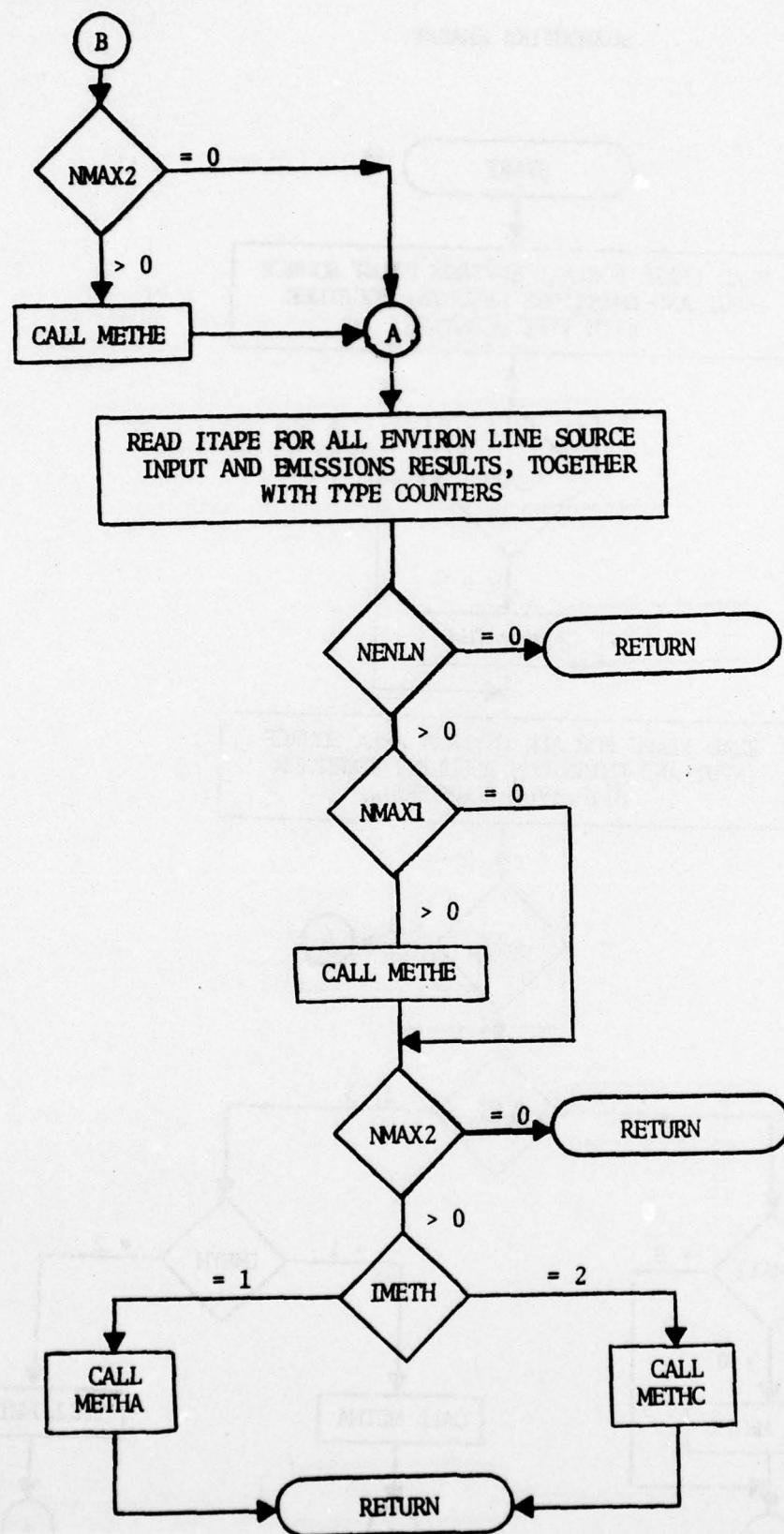
The arrays, ENPT, ENAR, and ENLN, are filled with geometry and emission data for all environ sources.

### Subroutines Called:

METHA, METHB, METHC, METHE

SUBROUTINE ENARAY





```

SUBFCUTINE ENAPAY ENARY000
C ENARY001
C THIS ROUTINE COMPUTES THE EMISSION RATES FOR ALL ENARY002
C ENVIFCN SOURCES ENARY003
C ENARY004
C CCOMMON / DEFAUT / ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7) ENARY005
CCMCMN/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200) ENARY006
. ,LOC1,LOC2,NGEOM,IPT ENARY007
CCMCMN/MONMET/TMBAR,WSMBAR,AMDMBR,DTMBAR FNARY008
COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD, ENARY009
. JSTAEC,HLID,TEMF,TEMK ENARY010
COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG ENARY011
CCMCMN/DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13), ENARY012
. VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13), ENARY013
. CVENDY(2),CVENHR(24),FLMC(13,7),FLDY(2,7),FLHR(24,7),NC1 ENARY014
CCMCMN /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAEC,NABLN, ENARY015
. NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20), ENARY016
. ABPT(16,150),ABAR(11,100),ABLN(14,100) ENARY017
C ENARY018
C*****POINTS ENARY019
C ENARY020
C READ(ITAPE) NFNPT,NTOT,((SORCE(I,N),I=1,NTOT),N=1,NENPT) ENARY021
IF (NENPT.EQ.0) GO TO 100 ENARY022
ICLASS=201 ENARY023
LCC1=10 ENARY024
LCC2=11 ENARY025
NGEOM=9 ENARY026
I1=16 ENARY027
I2=100 ENARY028
IPT=1 ENARY029
NSRCE=0 ENARY030
CALL METHA(NENPT,ENPT,I1,I2,ICLASS) ENARY031
C FNARY032
C*****AREAS ENARY033
C ENARY034
100 READ(ITAPE) NENAR,NTOT,IOPT,NMAX1,NMAX2, ENARY035
1 ((SCRCE(I,N),I=1,NTCT),N=1,NENAR) ENARY036
IF (NENAR.EQ.0) GO TO 300 ENARY037
LCC1=5 ENARY038
LCC2=7 ENARY039
NGECM=5 ENARY040
IFT=0 ENARY041
I1=11 ENARY042
I2=100 ENARY043
NSRCE=0 ENARY044
GC TC (110,120,130),IOPT ENARY045
C ENARY046
C*****OPTION 1 STATIONARY AREAS ENARY047
C FNARY048
110 ICLASS=202 ENARY049
IF (NMAX1.GT.0) ENARY050
1 CALL METHB(NMAX1,ENAR,I1,I2,ICLASS) ENARY051
C FNARY052
C*****OPTION 1 MOBILE AREAS ENARY053
C ENARY054
IF (NMAX2.GT.0) ENARY055
1 CALL METHE(NMAX2,ENAR,CVENMO,CVENDY,CVENHR,I1,I2) ENARY056
GC TC 300 ENARY057
C ENARY058
C*****OPTION 2 OF 3 LAND USE OF COMBINED AREAS ENARY059
C ENARY060
120 ICLASS=203 ENARY061

```

```

GC TC 200
130 ICCLASS=204
200 IMETH=1
IF (JFLAG.EQ.0) READ 201,IMETH
201 FCRMAT(I4)
IF (IMETH.EQ.1) CALL METHA(NMAX1,ENAR,I1,I2,ICCLASS)
IF (IMETH.EQ.2) CALL METHC(NMAX1,ENAR,I1,I2,ICCLASS)
C
C*****LINES
C      NMAX1 = NO. OF ROADWAY LINES
C      NMAX2 = NO. OF NON-ROADWAY LINES
C
300 READ(ITAPE) NENLN,NTOT,NMAX1,NMAX2,
1  ((SORCE(I,N),I=1,NTOT),N=1,NENLN)
IF (NENLN.EQ.0) GO TO 400
LCC1=8
LCC2=10
NGEOM=8
NSRCE=0
I1=14
I2=20
IET=0
IF (NMAX1.GT.0)
1 CALL METHE(NMAX1,ENLN,CVENMO,CVENDY,CVENHR,I1,I2)
C
IF (NMAX2.EQ.0) GO TO 400
ICCLASS=206
IMETH=1
IF (JFLAG.EQ.0) READ 201,IMETH
IF (IMETH.EQ.1) CALL METHA(NMAX2,ENLN,I1,I2,ICCLASS)
IF (IMETH.EQ.2) CALL METHC(NMAX2,ENLN,I1,I2,ICCLASS)
400 RETURN
END

```

ENARY062  
ENARY063  
ENARY064  
ENARY065  
ENARY066  
ENARY067  
ENARY068  
ENARY069  
ENARY070  
ENARY071  
ENARY072  
ENARY073  
ENARY074  
ENARY075  
ENARY076  
ENARY077  
ENARY078  
ENARY079  
ENARY080  
ENARY081  
ENARY082  
ENARY083  
ENARY084  
ENARY085  
ENARY086  
ENARY087  
ENARY088  
ENARY089  
ENARY090  
ENARY091  
ENARY092  
ENARY093  
ENARY094

SUBROUTINE INDINP  
ENTRY DEPINP

Purpose:

To print the input parameters for both wind independent and wind dependent sources.

Input:

All source parameters.

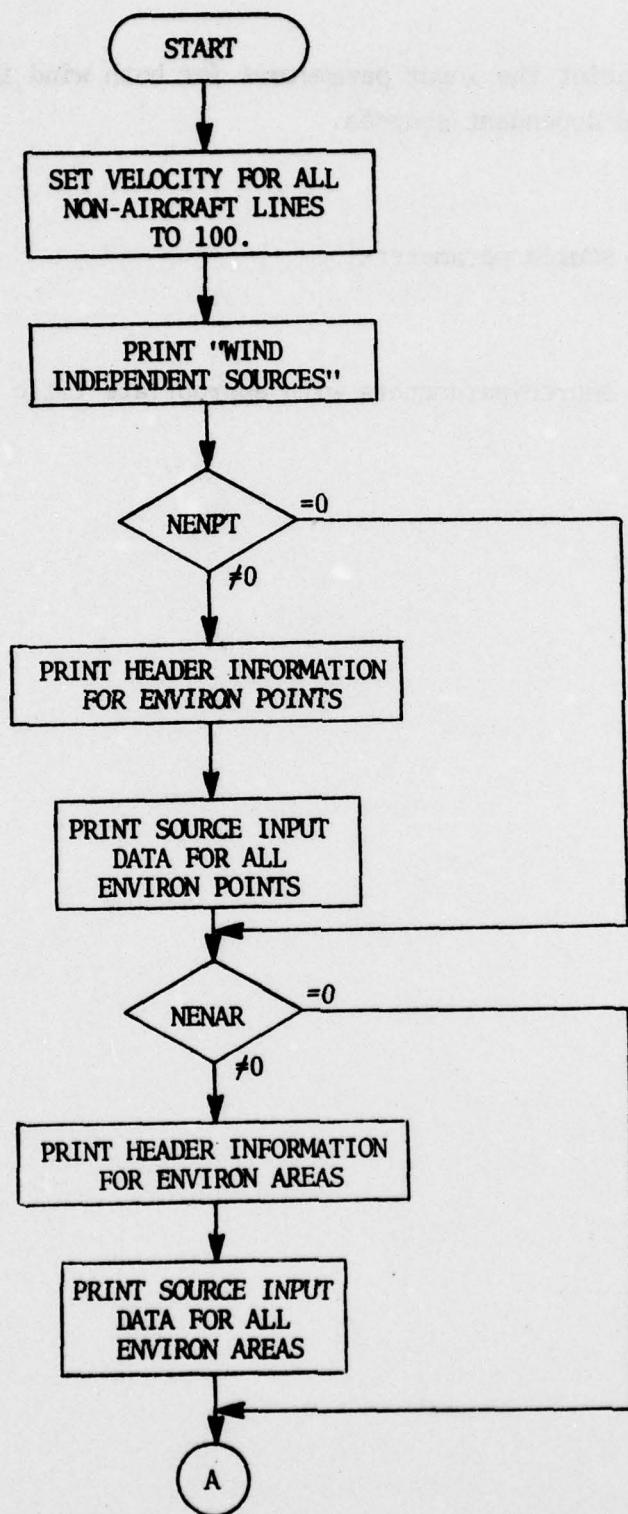
Output:

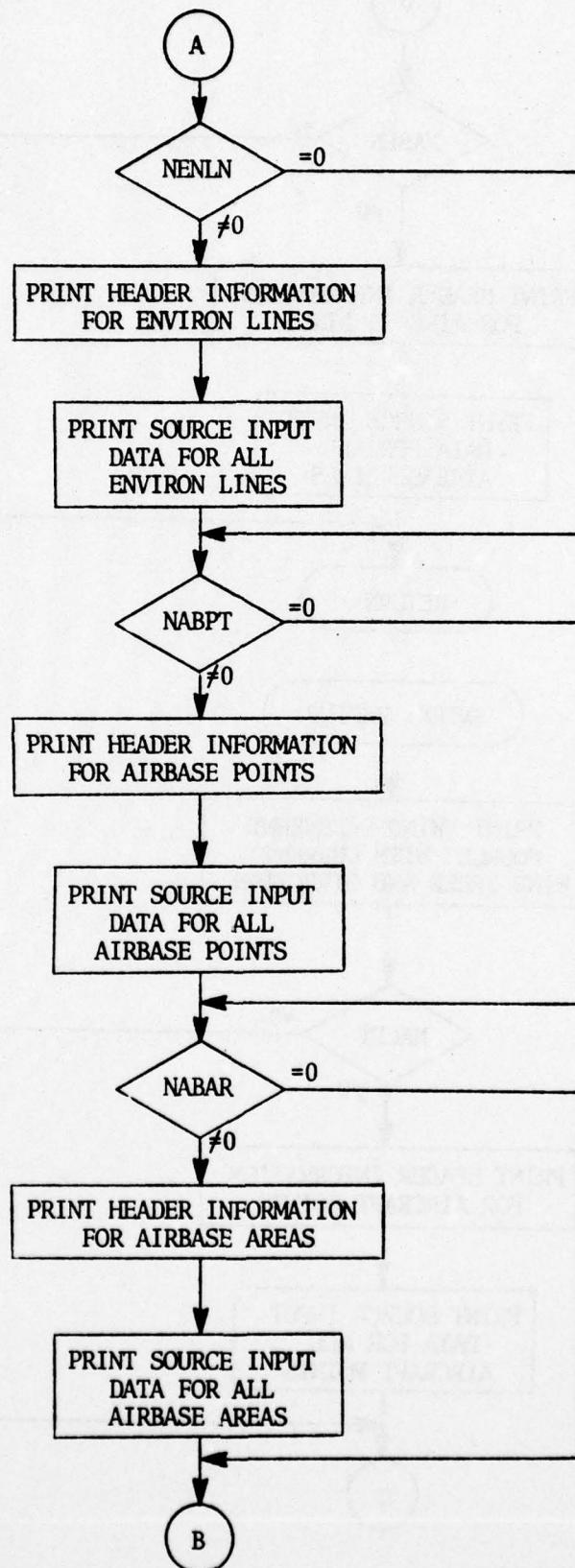
All source parameters with appropriate title information.

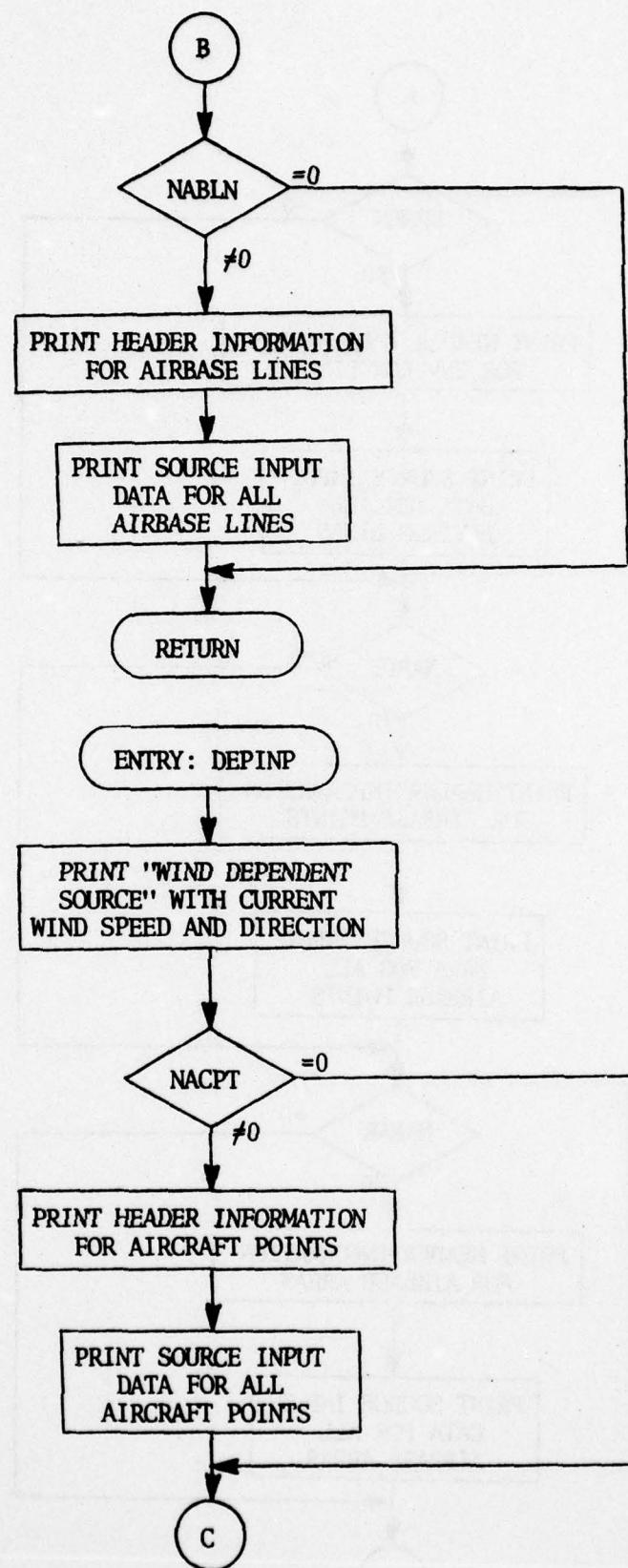
Subroutines  
Called:

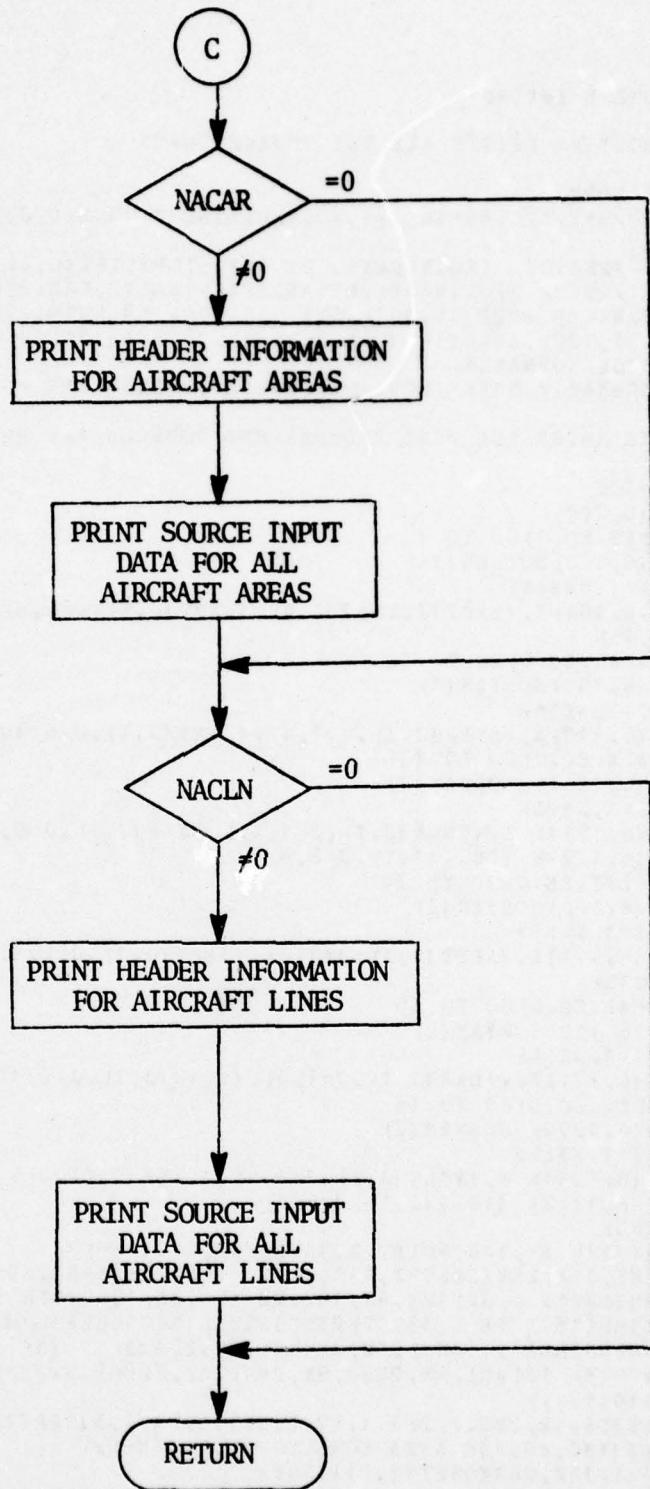
None

SUBROUTINE INDINP  
(ENTRY: DEPINP)









```

SUBROUTINE INDINP
C
C THIS ROUTINE PRINTS ALL THE SOURCE INPUT
C
REAL*8 SORNAME
CCMMCN /MET/ WS,WSMEH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF
,TEMK
COMMON /PERIOD/ IMO,NCDAYS,IDX,IHR1,IHR2,IFLAG,JFLAG
COMMON /SRCE/ NPOL,NENFT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
, NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
, ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
DIMENSION SORNAME(3)
DATA SCRNAME / 8HENIRON ,8HAIRBASE ,8HAIRCRAFT /
C
C AT THIS ENTRY ALL WINE INDEPENDENT SOURCES ARE PRINTED
C
ENLN=1.0
WRITE(6,200)
IF(NENFT.EQ.0)GO TO 11
WRITE(6,100)SORNAME(1)
DC 1 I=1,NENPT
1 WRITE(6,101)I,(ENPT(J,I),J=1,4),(ENPT(J,I),J=6,8),(ENPT(J,I),
,J=10,15)
11 IF(NENAR.EQ.0)GO TO 12
WRITE(6,110)SORNAME(1)
DC 2 I=1,NENAR
2 WRITE(6,111)I,(ENAR(J,I),J=1,4),(ENAR(J,I),J=6,10)
12 IF(NENLN.EQ.0)GO TO 13
WRITE(6,1200)SORNAME(1)
DO 5 I=1,NENLN
WRITE(6,1211) I,(ENLN(J,I),J=1,4),(ENLN(J,I),J=9,13)
5 WRITE(6,1222) (ENLN(J,I),J=6,8)
13 IF(NABPT.EQ.0)GO TO 14
WRITE(6,100)SORNAME(2)
DO 3 I=1,NABPT
3 WRITE(6,101)I,(ABPT(J,I),J=1,4),(ABPT(J,I),J=6,8),(ABPT(J,I),
,J=10,15)
14 IF(NABAR.EQ.0)GO TO 15
WRITE(6,110)SORNAME(2)
DO 4 I=1,NABAR
4 WRITE(6,111)I,(ABAR(J,I),J=1,4),(ABAR(J,I),J=6,10)
15 IF(NABLN.EQ.0)GO TO 16
WRITE(6,1200)SORNAME(2)
DC 6 I=1,NAELN
WRITE(6,1211) I,(ABLN(J,I),J=1,4),(ABLN(J,I),J=9,13)
6 WRITE(6,1222) (ABLN(J,I),J=6,8)
16 CONTINUE
100 FCRRMAT(1H0,A8,14H POINT SOURCES/1X,119(1H-)/
, 8X,1HI,11X,8HGEOMETRY,11X,22HI STACK PARAMETERS I,4X,1HI/
, 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,3X,12HWIDTH I TEMP,4X,3HVEL,
, 3X,11HDIAM I PR I,13X,28HEMISSIONS(MICROGRAMS/SECOND)/
, 1X,14HNUMBER I (KM),5X,4H(KM),4X,25H(M) (M) I (DEG K) (M/S),
, 3X,10H(M) IFLAGI,4X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/
, 1X,119(1H-))
101 FORMAT(I6,1X,2F9.2,2F7.1,F7.0,2F7.1,F4.0,5(1PE11.3))
110 FCRRMAT(1H0,A8,13H ARFA SOURCES/1X,94(1H-)/
, 8X,1HI,11X,8HGEOMETRY,11X,1HI/
, 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,4X,6HSIDE I,
, 14X,28HEMISSIONS(MICROGRAMS/SECOND)/
, 1X,14HNUMBER I (KM),5X,4H(KM),4X,3H(M),4X,5H(M) I,
, 5X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/1X,94(1H-))
111 FCRRMAT(I6,1X,2F9.2,2F7.1,5(1PE11.3))

```

```

120 FORMAT(1H0,A8,13H LINE SOURCES/1X,123(1H-)/
. 8X,1HI,11X,8HGEOMETRY,11X,1HI,10X,1HI,54X,15HI AIRCRAFT ONLY/
. 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,3X,18HWIDTH I VELOCITY I,
. 13X,28HEMISSIONS(MICROGRAMS/SECOND),13X,15HI LENGTH TIME/
. 1X,14HNUMBER I (KM),5X,4H(KM),4X,23H(M) (M) I (KM/HR) I,
. 5X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2,3X,15HI (KM) (HR)/
. 1X,123(1H-))
121 FORMAT(I6,1X,2F9.2,2F7.1,6(1PE11.3),0PF7.2,1PE11.3) INDIP062
122 FCRRMAT(7X,2F9.2,F7.1,7X,1PE11.3) INDIP063
1200 FCRRMAT(1H0,A8,13H LINE SOURCES/1X,96(1H-)/
. 8X,1HI,11X,8HGEOMETRY,12X,1HI/ INDIP064
. 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,4X,7HWIDTH I, INDIP065
. 23X,28HEMISSIONS(MICROGRAMS/SECOND)/ INDIP066
. 1X,14HNUMBER I (KM),5X,4H(KM),4X,3H(M),4X,6H(M) I, INDIP067
. 6X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/ INDIP068
. 1X,96(1H-)) INDIP069
1211 FORMAT(I6,1X,2F9.2,2F7.1,2X,5(1PE11.3)) INDIP070
1222 FCRRMAT(7X,2F9.2,F7.1) INDIP071
200 FORMAT(25HOWIND INDEPENDENT SOURCES/1H0) INDIP072
      RETUFN INDIP073
C      ENTRY DEPIMP INDIP074
C      AT THIS ENTRY ALL WIND DEPENDENT SOURCES ARE PRINTED INDIP075
C      WRITE(6,300) WS,WD INDIP076
300 FCRRMAT(1H1,'WIND DEPENDENT SOURCES FOR',F8.4,' MPS WIND SPEED AND'INDIP077
. ,F8.4,' RADIANS WIND DIRECTION')
IF (IFLAG.EQ.0) GO TO 18 INDIP078
IF (NACPT.EQ.0) GO TO 17 INDIP079
WRITE(6,100) SORNAME(3) INDIP080
DC 7 I=1,NACPT INDIP081
7 WPITE(6,101) I,(ACPT(J,I),J=1,4),(ACPT(J,I),J=6,8),(ACPT(J,I),
. J=11,15) INDIP082
17 IF(NACAR.EQ.0) GO TO 18 INDIP083
WRITE(6,110) SORNAME(3) INDIP084
DC 8 I=1,NACAR INDIP085
8 WRITE(6,111) I,(ACAR(J,I),J=1,4),(ACAR(J,I),J=6,10) INDIP086
18 IF(NACIN.EQ.0) GO TO 19 INDIP087
WRITE(6,120) SORNAME(3) INDIP088
DC 9 I=1,NACIN INDIP089
IF (ACIN(9,I).NE.1.0) GO TO 1987 INDIP090
WRITE(6,1219) I,(ACLN(J,I),J=1,4),(ACLN(J,I),J=13,17),ACLN(11,I) INDIP091
WEITE(6,1229) (ACLN(J,I),J=6,8) INDIP092
1219 FORMAT(16,1X,2F9.2,2F7.1,4X,3HN/A,4X,5(1PE11.3),0PF7.1,4X,3HN/A) INDIP093
1229 FORMAT(7X,2F9.2,F7.1,11X,3HN/A) INDIP094
      GC TC 9 INDIP095
1987 CONTINUE INDIP096
      WRITE(6,121) I,(ACLN(J,I),J=1,4),ACLN(9,I),(ACLN(J,I),J=13,17),
. ACLN(11,I),ACLN(12,I) INDIP097
      WRITE(6,122) (ACLN(J,I),J=6,8),ACLN(10,I) INDIP098
9 CCNTINUE INDIP099
19 CONTINUE INDIP100
      RETUFN INDIP101
      END INDIP102

```

## PROGRAM MAIN

### Purpose:

To read the general problem input, set up the receptor grid, call a routine to read the master emission file and then call the short-term model.

### Input:

1. Problem title
2. Definition of pollutants to be output
3. Description of special cases
4. Description of receptor grid
5. Description of statistical receptors

### Output:

All input is printed

### Procedure:

1. Read card input
2. Calculate receptor locations
3. Check statistical receptors against the receptor locations
4. Call routine to read master emission file
5. Call the short-term model

### Subroutines

#### Called:

READ, MAINS

AD-A046 348

ARGONNE NATIONAL LAB ILL

AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-T--ETC(U)

APR 77 D J BINGAMAN

F/G 13/2

UNCLASSIFIED

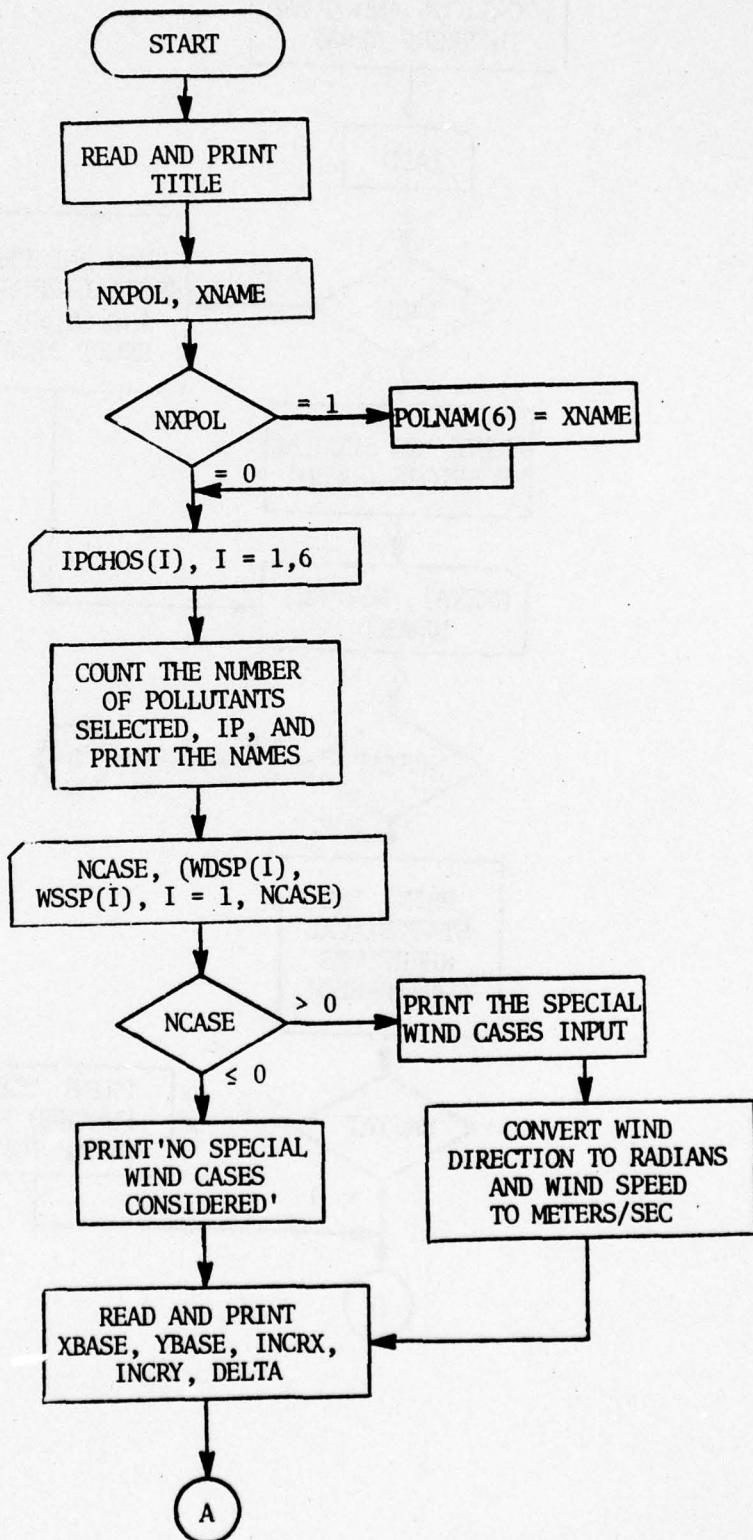
CEEDO-TR-76-34

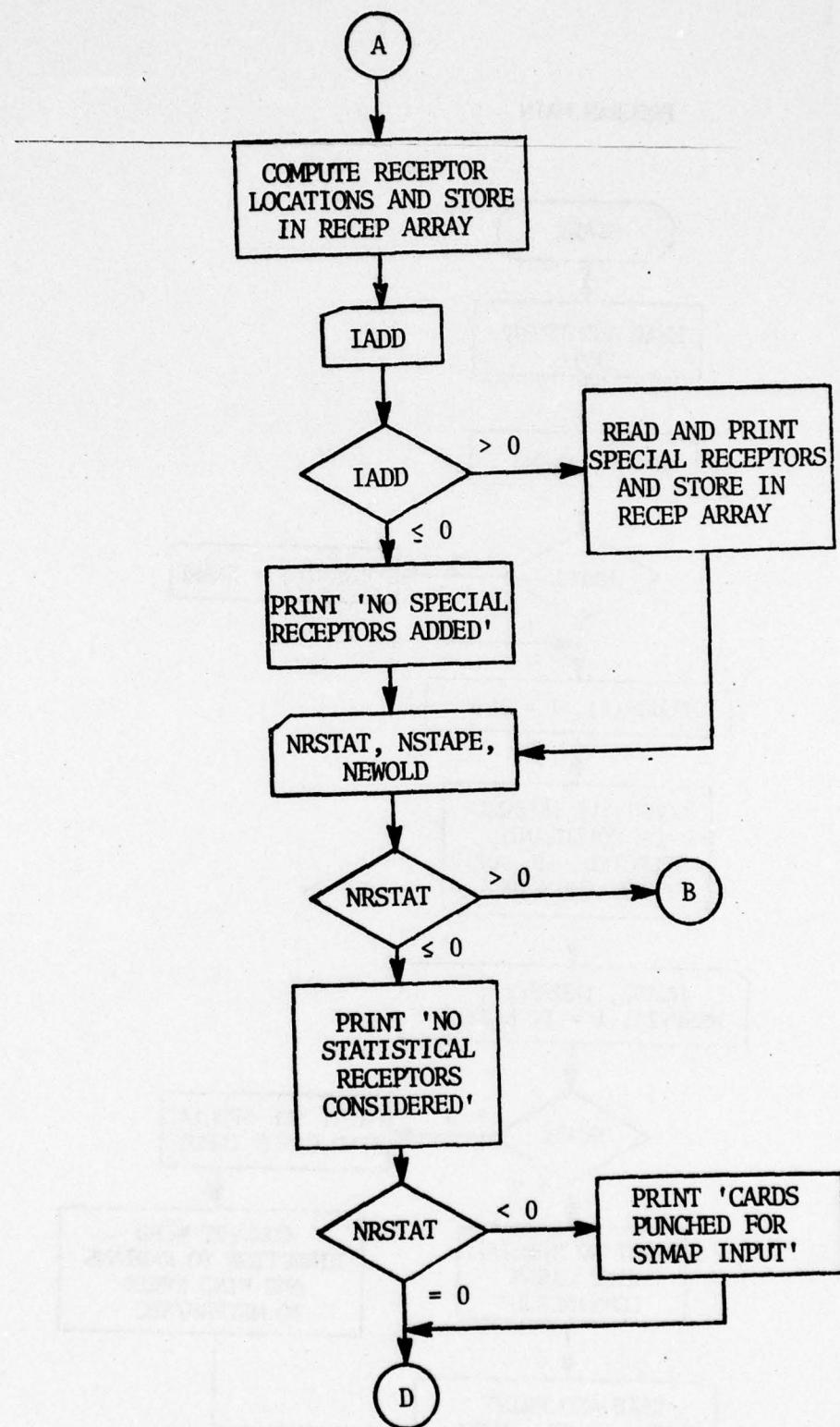
NL

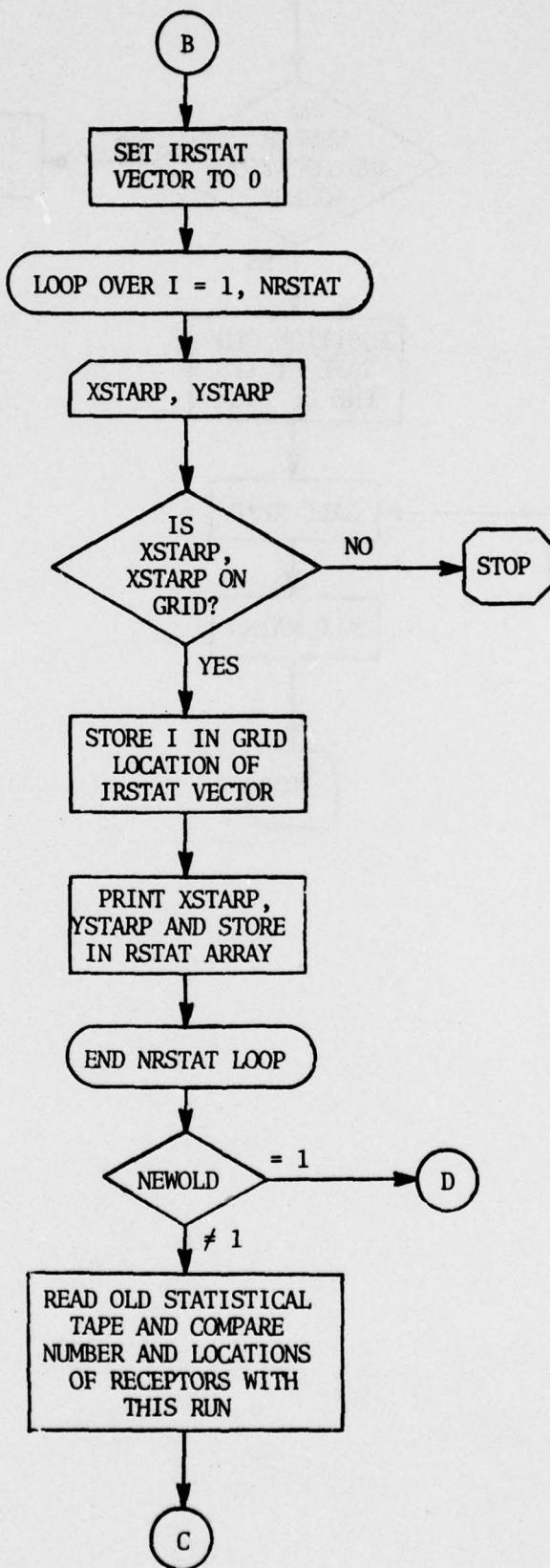
2 OF 3  
AD  
A046348

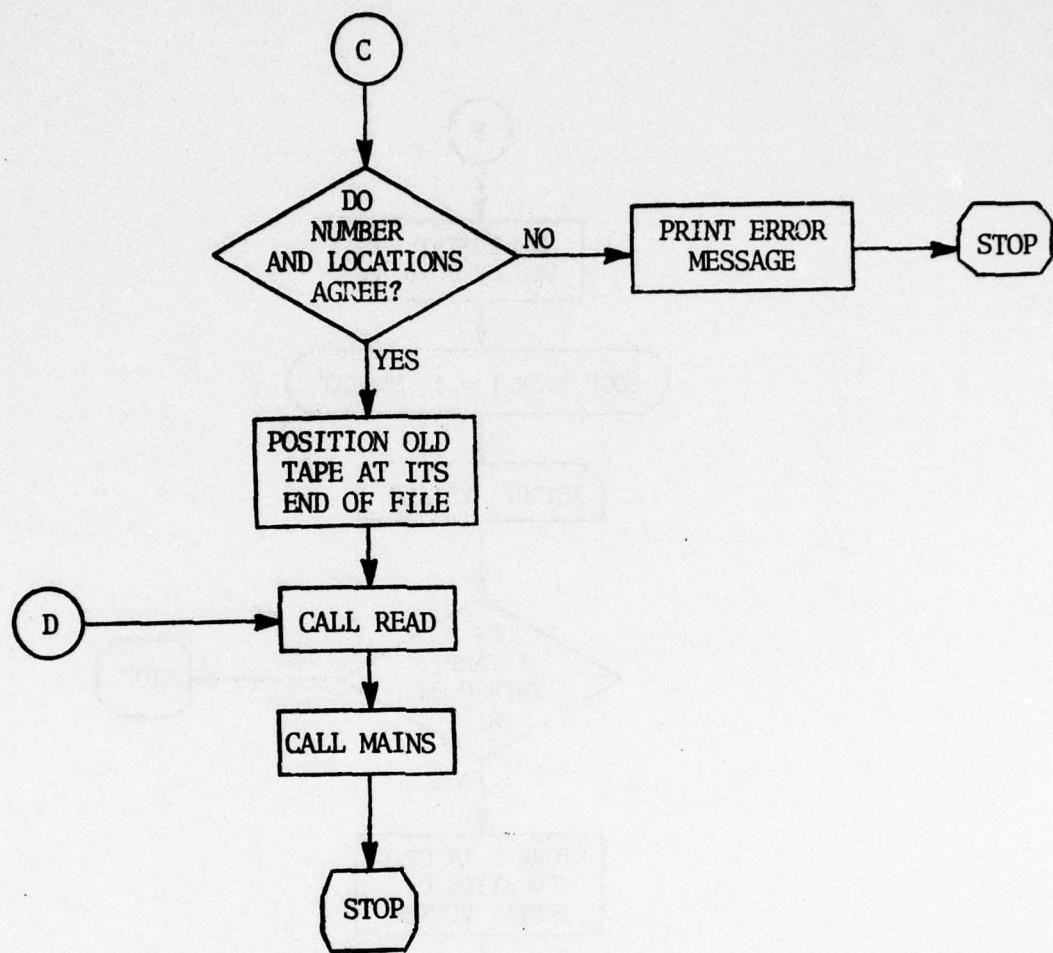


PROGRAM MAIN









```

C THIS PROGRAM IS THE MAIN DRIVER ROUTINE WHICH READS IN RECEPTOR AND MAIN000C
C OTHER GENERAL DATA, CALLS SUBROUTINE READ TO READ THE MASTER MAIN0001
C SOURCE EMISSION TAPE, AND THE DIRECTS CONTROL TO MAINS FOR THE MAIN0002
C SHORT TERM MODEL MAIN0003
C MAIN0004
C MAIN0005
C MAIN0006
C MAIN0007
C MAIN0008
C MAIN0009
C MAIN0010
C MAIN0011
C MAIN0012
C MAIN0013
C MAIN0014
C MAIN0015
C MAIN0016
C MAIN0017
C MAIN0018
C MAIN0019
C MAIN0020
C MAIN0021
C MAIN0022
C MAIN0023
C MAIN0024
C MAIN0025
C MAIN0026
C MAIN0027
C MAIN0028
C MAIN0029
C MAIN0030
C MAIN0031
C MAIN0032
C MAIN0033
C MAIN0034
C MAIN0035
C MAIN0036
C MAIN0037
C MAIN0038
C MAIN0039
C MAIN0040
C MAIN0041
C MAIN0042
C MAIN0043
C MAIN0044
C MAIN0045
C MAIN0046
C MAIN0047
C MAIN0048
C MAIN0049
C MAIN0050
C MAIN0051
C MAIN0052
C MAIN0053
C MAIN0054
C MAIN0055
C MAIN0056
C MAIN0057
C MAIN0058
C MAIN0059
C MAIN0060
C MAIN0061
REAL*8 POLNAM,XNAME
COMMON /AIRQAL/ RECDAT(3, 6, 312)
COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR
COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,SORC(18),IPOL
COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP
      ,TEMK,UA
COMMON /MCNMET/ TMBAR,WSMBAR,AMDMBR,DTMBAR
COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG,IONCE
COMMON /RCPT/ NRECEP,RECEF(2, 312)
COMMON /SPEC/ NCASE,WSSP(3),WDSP(3)
COMMON /SRCE/ NPOL,NEKPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
      . NACAR,NACLN,ENPT(16, 100),ENAR(11, 100),ENLN(14, 20),ABPT(16, 150),
      . ABAR(11, 100),ABLN(14, 100),ACPT(16, 1),ACAR(11, 24),ACLN(18, 250)
COMMON /TITLE/ POLNAM(6),TITLE1(20),IPCHOS(6),NXPOL,IP
COMMON /STAT/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)
DIMENSION AA (20),OSTATR(2,20)

C READ AND PRINT RECEPTOR AND OTHER GENERAL INPUT
C
1 READ(5,100) TITLE1
100 FCRMAT(20A4)
      PRINT 200, TITLE1
200 FORMAT(1H1,20A4)
      READ(5,110)      NXPOL, XNAME
110 FCRMAT( 16,5A8)
      IF(NXFOL.EQ.0) GO TO 31
      FCLNAM(6 )=XNAME
31 CONTINUE
      READ(5,130) (IPCHOS(I),I=1,6)
130 FCRMAT(10I6)
      DO 40 I=1,6
      IF(IPCHOS(I).LE.0) GO TO 41
40 CONTINUE
41 IE=I-1
      PRINT 203, (POLNAM(IPCHOS(I)),I=1,IP)
203 FORMAT(21H0 POLLUTANTS SELECTED /6A8)
      READ(5,140) NCASE, (WDSP(I),WSSP(I),I=1,NCASE)
140 FORMAT(I6,6F6.0)
      IF (NCASE) 48,48,49
48 PRINT 201
201 FORMAT(33H0 NO SPECIAL WIND CASES CONSIDERED)
      GC TC 51
49 PRINT 202,      (I,WDSP(I),WSSP(I),I=1,NCASE)
202 FCRMAT(20H0 SPECIAL WIND CASES /53H      CASE      WIND DIRECTION(DEGREES
      .)      WIND SPEED(KNOTS)/(I6,F18.2,F23.2))
      DC 50 I=1,NCASE
      WDSP(I)=WDSP(I)* 0.0174533
50 WSSP(I)=WSSP(I)*0.5148
51 CCNTINUE
      READ(5,120) XBASE,YBASE,INCRX,INCRY,DELTA
120 FORMAT(2F8.0,2I8,F8.0)
      PRINT 204, XBASE,YBASE,INCRX,INCRY,DELTA
204 FORMAT(43H0 LOWER LEFT CORNER OF RECEPTOR GRID IS AT (,F8.3,1H,, F8.3,1H)/12H
      . THERE ARE,I4,12H COLUMNS AND,I4,23H ROWS WITH A SPACED
      . CING OF,F6.2,11H KILOMETERS)
      NRECEF=0

```

```

DG 10 I=1,INCPX          MAIN0062
DC 10 J=1,INCRY          MAIN0063
NRECEP=NRECEP+1          MAIN0064
RECEP(1,NRECEP)=XBASE+(I-1)*DELTA MAIN0065
10 RECEP(2,NRECEP)=YBASE+(J-1)*DELTA MAIN0066
READ(5,110)IADD          MAIN0067
IF(IADD)14,14,15          MAIN0068
14 PRINT 205              MAIN0069
205 FORMAT(27HONO SPECIAL RECEPPIORS ADDED) MAIN0070
GC TC 21                 MAIN0071
15 PRINT 206              MAIN0072
206 FORMAT(25HOSPECIAL RECEPPIORS ADDED /36H NO. X-COORDINATE Y-CMAINC073
.OCRDINATE)
DO 20 I=1,IADD          MAIN0074
READ(5,120)XRECEP,YRECEP MAIN0075
NRECEP=NRECEP+1          MAIN0076
PRINT 207, NRECEP,XRECEP,YRECEP MAIN0077
207 FORMAT(I5,F14.3,F15.3) MAIN0078
RECEP(1,NRECEP)=XRECEP MAIN0079
RECEP(2,NRECEP)=YRECEP MAIN0080
20 CCNTINUE              MAIN0081
21 CONTINUE               MAIN0082
C
C READ AND PRINT STATISTICAL RECEPTOR INPUT
C
READ 130,NRSTAT,NSTAPE,NEWOLD          MAIN0083
IF (NRSTAT.GT.0) GO TC 305          MAIN0084
PRINT 302              MAIN0085
302 FORMAT(36HONO STATISTICAL RECEPPIORS CONSIDERED) MAIN0086
IF (NRSTAT.LT.0) PRINT 303          MAIN0087
303 FCRMAT(30H0CAPDS PUNCHED FOR SYMAP INPUT) MAIN0088
GO TC 400              MAIN0089
305 PRINT 301,NRSTAT          MAIN0090
301 FORMAT(1H0,I8,22H STATISTICAL RECEPPIORS) MAIN0091
DO 310 I=1,NRECEP          MAIN0092
IRSTAT(I)=0              MAIN0093
310 CCNTINUE              MAIN0094
DC 340 I=1,NRSTAT          MAIN0095
READ 120,XSTARP,YSTARP          MAIN0096
DO 320 IC=1,NRECEP          MAIN0097
IF (XSTARP.EQ.RECEP(1,IC).AND.YSTARP.EQ.RECEP(2,IC)) GO TO 330 MAIN0098
320 CCNTINUE              MAIN0099
PRINT 321,XSTARP,YSTARP          MAIN0100
321 FORMAT(25H0STATISTICAL RECEPTOR X =,F7.3,5H, Y =,F7.3, MAIN0101
. 12H NOT ON GRID)          MAIN0102
STOP
330 IRSTAT(IC)=I          MAIN0103
PRINT 322,I,XSTARP,YSTARP          MAIN0104
322 FORMAT(I12,7H AT X =,F10.3,4H Y =,F10.3) MAIN0105
RSTAT(1,I)=XSTARP          MAIN0106
RSTAT(2,I)=YSTARP          MAIN0107
340 CCNTINUE              MAIN0108
IF (NEWOLD.EQ.1) GO TO 400          MAIN0109
READ (NSTAPE) AA,IH,JRSTAT,((OSTATR(I,J),I=1,2),J=1,JRSTAT), MAIN0110
. JWE,WDJ,JWS,WSJ,JJJ,HLIDJ          MAIN0111
IF (JRSTAT.EQ.NRSTAT) GO TO 350          MAIN0112
PRINT 341              MAIN0113
341 FORMAT(46HONUMBER OF STATISTICAL RECEPPIORS ON OLD TAPE (,I2, MAIN0114
. 42H), DO NOT AGREE WITH NUMBER FOR THIS RUN (,I2,1H))          MAIN0115
STOP
350 DC 360 J=1,NRSTAT          MAIN0116
DC 360 I=1,2              MAIN0117
                                         MAIN0118
                                         MAIN0119
                                         MAIN0120
                                         MAIN0121
                                         MAIN0122
                                         MAIN0123

```

```
IF (CSTATR(I,J) .EQ. RSTAT(I,J)) GO TO 360 MAIN0124
PRINT 351,((CSTATR(I,J),I=1,2),J=1,NRSTAT) MAIN0125
351 FCRMAT(64H0STATISTICAL RECEPTOR LOCATIONS DO NOT MATCH,THOSE ON TA MAIN0126
.PE ARE ,20(/2F15.3)) MAIN0127
STOP MAIN0128
360 CONTINUE MAIN0129
C MAIN0130
C POSITION OLD TAPE AT END OF LAST RECORD MAIN0131
C MAIN0132
370 READ (NSTAPE,END=380) MAIN0133
GO TO 370 MAIN0134
380 BACKSPACE NSTAPE MAIN0135
BACKSPACE NSTAPE MAIN0136
400 CCNTINUE MAIN0137
CALL READ MAIN0138
CALL MAINS MAIN0139
STOP MAIN0140
RETURN MAIN0141
END MAIN0142
```

## SUBROUTINE MAINS

### Purpose:

To direct the short term calculation by reading the data, checking for special cases and calling the source and diffusion calculation routines.

### Input:

1. Card input to describe the time periods to be calculated
2. Card input to describe the meteorological conditions
3. Special case data from the MAIN routine

### Output:

1. Common block data to be used by the calculation and output subroutines
2. Statistical receptor data on tape and/or cards for SYMAP

### Procedure:

1. Set constants
2. Read general data
3. For each period:
  - a. Read time and meteorological data
  - b. Check for near zero wind speed
  - c. Check limits of mixing depth
  - d. For each hour:
    1. Find the critical distance
    2. Set wind direction and speed classes
    3. Call the non-aircraft source routines
    4. Check for special cases of wind speed and direction
    5. Call the aircraft source routines
    6. Call the diffusion model and output routines
    7. Check for statistical output, including cards for SYMAP.

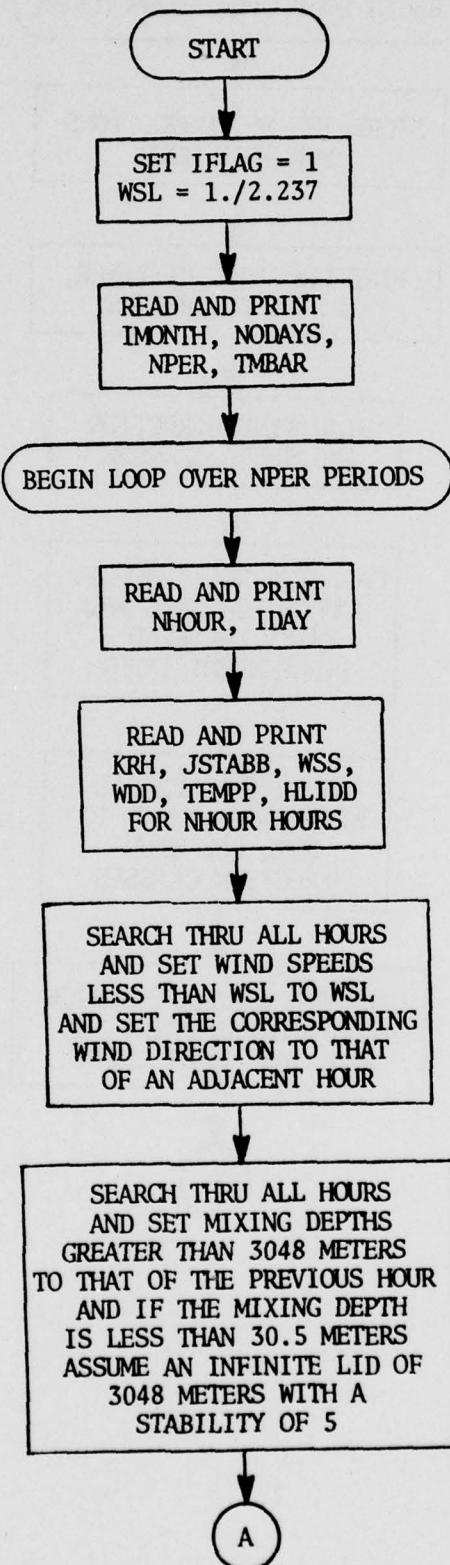
### Subroutines Called:

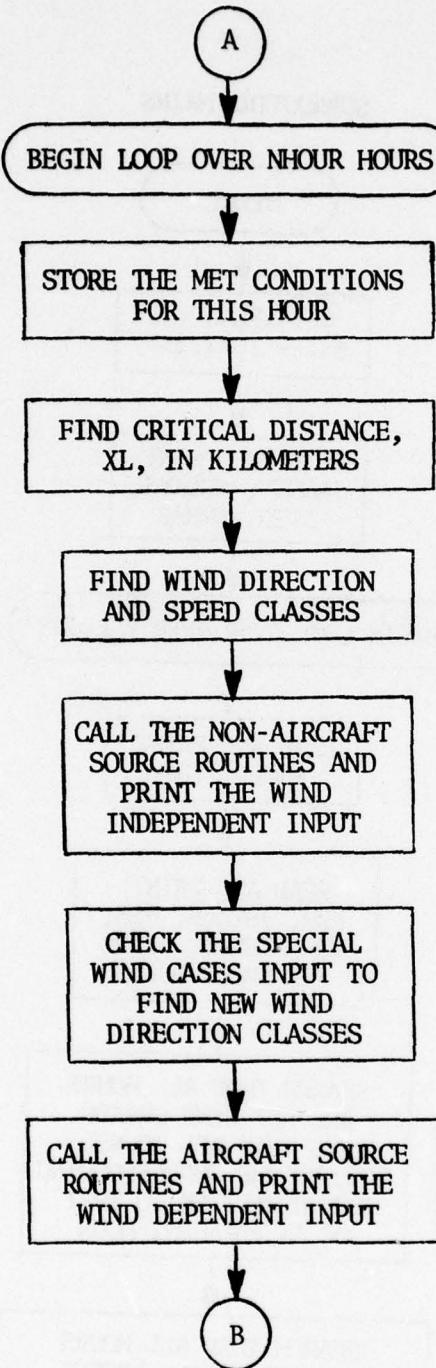
SOURCE, INDINP, ACSRCE, DEPINP, POLSOR, OUTPUT

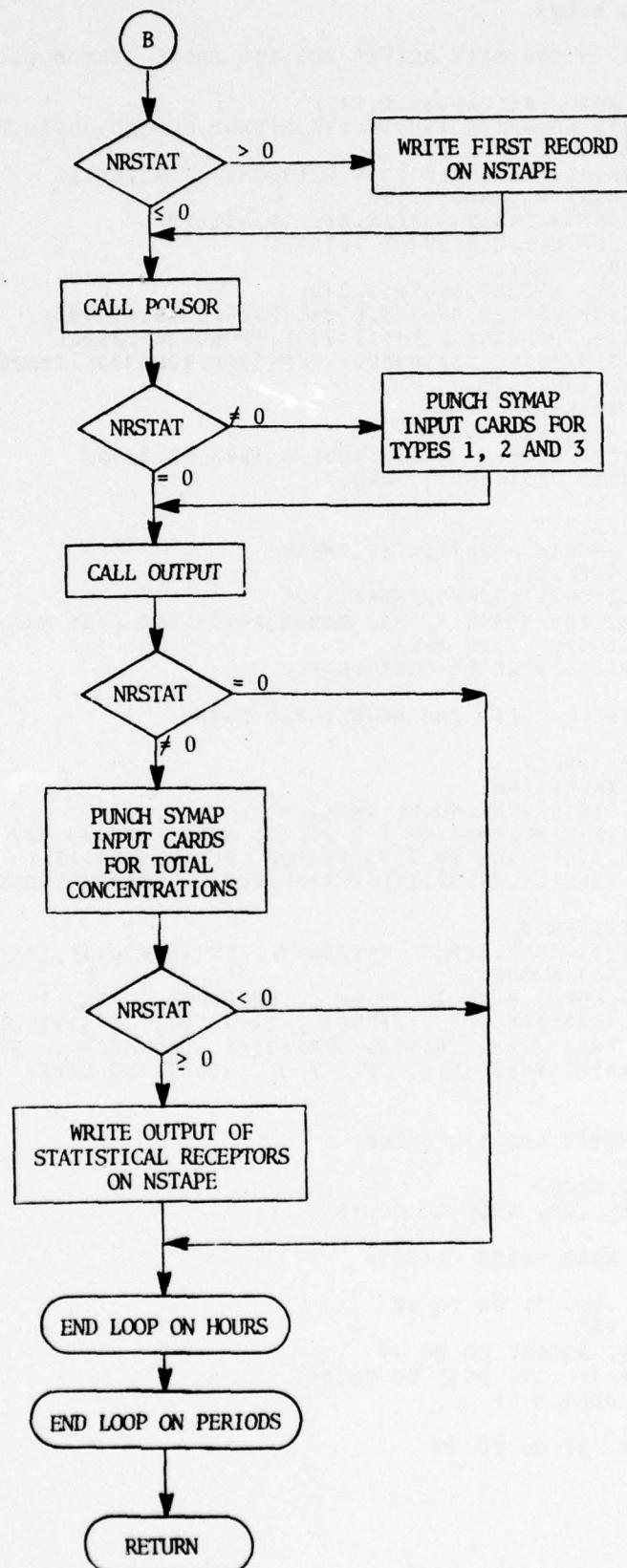
### Function Called:

SIGCZ

SUBROUTINE MAINS







```

SUBROUTINE MAINS                               MAINS000
C THIS ROUTINE IS THE MAIN DRIVER FOR THE SHORT TERM MODEL   MAINS001
C
COMMON /AIRQAL/ RECDAT(3, 6,312)             MAINS002
COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,   MAINS003
. TEMK                                         MAINS004
COMMON /PERIOD/ IMONTH,NODAYS,IDAD,IHR1,IHR2,IFLAG             MAINS005
CMMCN /XTRAN/ XL,WSMD,TY,TZ                  MAINS006
COMMON /MONMET/ TMBAR,WSMBAR,AMDMBR,DTMBAR             MAINS007
CMMCN /SPEC/ NCASE,WSSP(3),WDSP(3)             MAINS008
COMMON /WDUN/ WSAVE                           MAINS009
COMMNCN /RCPT/ NRECEP, RECEP(2,312)           MAINS010
COMMON /STAT/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)          MAINS011
COMMNCN /TITLE/ PCLNAM(6),TITLE1(20),IPCHOS(6),NXPOL          MAINS012
DIMENSION KRH(100),JSTABB(100),WSS(100),WDD(100),TEMPP(100),   MAINS013
. HLIDD(100),WSCLAS(5)                         MAINS014
DIMENSION WEEK(2)                           MAINS015
DATA WEEK / 3HDAY, 3HEND /                  MAINS016
DATA WSCLAS/1.8018,3.3462,5.4055,8.4943,10.8110/          MAINS017
DATA PI,SHLID /3.1415927,3048./                 MAINS018
IFLAG = 1                                     MAINS019
WSL = 1./2.237                                MAINS020
READ 100, IMONTH,NODAYS,NPER,TMBAR           MAINS021
100 FCRMAT (3I6,F6.0)                         MAINS022
PRINT 203,IMONTH,NODAYS,TMBAR,NPER           MAINS023
203 FCRMAT (11H0 FOR MONTH13,11H, THERE ARE14,36H DAYS WIH AN AVERAGE   MAINS024
. TEMPERATURE OFF6.2,4H (F)/                  MAINS025
.15,27H PERIODS WILL BE CONSIDERED)        MAINS026
C FOR EVERY PERIOD, READ THE HOURLY MET DATA   MAINS027
C
DO 50 IPER=1,NPER                         MAINS028
READ 100, NHOUR,IDAD                      MAINS029
PRINT 201, IPER,NPER,NHOUR,WEEK(IDAY)        MAINS030
201 FCRMAT (1H1,29HINFORMATION FOR PERIOD NUMBER,13,3H OF, 13,8H PERIOD   MAINS031
. S / 5X,13,33H HOURS IN THIS PERIOD, FOR A WEEK,A3 )          MAINS032
READ 101, (KRH(I),JSTABB(I),WSS(I),WDD(I),TEMPP(I),HLIDD(I),   MAINS033
-I=1,NHOUR)                                MAINS034
101 FORMAT (2I6,4F6.0)                      MAINS035
PRINT 202,(I,NHOUR,KRH(I),JSTABB(I),WSS(I),WDD(I),TEMPP(I),   MAINS036
. HLIDD(I),I=1,NHOUR)                         MAINS037
202 FORMAT (1H0,8HFOR HOUR,15,3H OF,15,6H HOURS /5X, 10HHOUR INDEX ,   MAINS038
. I3 / 5X, 18HSTABILITY CATEGORY , I2 / 5X, 26HWIND SPEED (METERS/MAIN
. SECOND) , F8.2 / 5X, 24HWIND DIRECTION (DEGREES) , F8.2 /   MAINS039
. 5X, 15HTEMPERATURE (F) , F8.2 / 5X, 21HMIXING DEPTH (METERS) ,   MAINS040
. F 8.2 )                                     MAINS041
C CHECK WIND SPEED AND LID HEIGHT          MAINS042
C
DO 20 IH=1,NHOUR                         MAINS043
IF (WSS(IH) .GE. WSL) GO TO 18           MAINS044
C REMOVE ZERO WIND SPEED CLASSES          MAINS045
C
IF (NHOUR .EQ. 1) GO TO 16               MAINS046
WSS(IH) = WSL                            MAINS047
IF (IH .EQ. NHOUR) GO TO 17             MAINS048
IF (WSS(IH+1) .LT. WSL) GO TO 14         MAINS049
WDD(IH) = WDD(IH+1)                      MAINS050
GO TO 18                                  MAINS051
14 IF (IH .NE. 1) GO TO 17             MAINS052

```

```

DC 15 JH=2, NHOUR
IF (WSS(JH) .LT. WSL) GO TO 15
WDD(IH) = WDD(JH)
GO TO 18
15 CCNTINUE
16 PRINT 200
STOF
200 FCRFORMAT(1H1,10X,'WINDSPEED IS TOO SMALL')
17 WDD(IH) = WDD(IH-1)

C      SET INFINITE LID HEIGHT AT 3048 METERS
C
18 IF(HLIDD(IH).GT. 3048.) HLIDD(IH) = SHLID
IF (HLIDD(IH) .GE. 30.5) GO TO 19
C      GROUND LEVEL INVERSION, ASSUME INFINITE LID WITH A STABILITY OF 5
C
C      HLIDD(IH) = 3048.
C      JSTABB(IH) = 5
19 SHLID = HLIDD(IH)
20 CCNTINUE

C      BEGIN HCURLY CALCULATIONS
C
DO 30 IHOUR = 1,NHOUR
IHR1 = KRH(IHOUR)
WS = WSS(IHOUR)
IHR2 = IHR1
WD = WDD(IHOUR)*0.0174533
TEMF = TEMPP(IHOUR)
TEMK = (TEMF-32.)/1.8 + 273.
JSTAB = JSTABB(IHOUR)
HLID = HLIDD(IHOUR)
WSMPH = WS * 2.237
SINEWD = SIN(WD)
COSEWD = COS(WD)
HL = 0.47*HLID
WSAVE=WS
IF(HL.LT.1.0) HL=1.0

C      FIND CRITICAL DISTANCE, XL, IN KILOMETERS
C
XL=SIGCZ(JSTAE,HL)/1000.

C      FIND WIND DIRECTION AND SPEED CLASSES
C
DC 21 K=1,16
IF (WD .GT. (22.5*K-11.25)*PI/180.) GO TO 21
IWD = K
GC TC 22
21 CCNTINUE
IWD=1
22 DC 23 K=1,5
IF (WS.GT.WSCLAS(K)) GO TO 23
IWS = K
GO TO 25
23 CCNTINUE
IWS = 6
25 CCNTINUE
PRINT 925,IHOUR
925 FORMAT(9H1FOR HOURI3)

C

```

MAINS062  
MAINS063  
MAINS064  
MAINS065  
MAINS066  
MAINS067  
MAINS068  
MAINS069  
MAINS070  
MAINS071  
MAINS072  
MAINS073  
MAINS074  
MAINS075  
MAINS076  
MAINS077  
MAINS078  
MAINS079  
MAINS080  
MAINS081  
MAINS082  
MAINS083  
MAINS084  
MAINS085  
MAINS086  
MAINS087  
MAINS088  
MAINS089  
MAINS090  
MAINS091  
MAINS092  
MAINS093  
MAINS094  
MAINS095  
MAINS096  
MAINS097  
MAINS098  
MAINS099  
MAINS100  
MAINS101  
MAINS102  
MAINS103  
MAINS104  
MAINS105  
MAINS106  
MAINS107  
MAINS108  
MAINS109  
MAINS110  
MAINS111  
MAINS112  
MAINS113  
MAINS114  
MAINS115  
MAINS116  
MAINS117  
MAINS118  
MAINS119  
MAINS120  
MAINS121  
MAINS122  
MAINS123

```

C CALL THE NCN-AIRCRAFT SOURCE ROUTINES AND PRINT THE          MAINS124
C WIND INDEPENDENT INPUT                                     MAINS125
C                                                       MAINS126
C
C CALL SCURCE                                              MAINS127
C CALL INDINP                                              MAINS128
C IWDT = IWD                                              MAINS129
C IF (NCASE .EQ. 0) GO TO 28                                MAINS130
C DC 26 I=1,NCASE                                         MAINS131
C COMP = WS * COS(WD - WDSP(I))                           MAINS132
C IF (COMP .LT. WSSP(I)) GO TO 26                           MAINS133
C IWDT = 17 + I                                           MAINS134
C GO TO 28                                              MAINS135
C 26 CCNTINUE                                              MAINS136
C 28 CCNTINUE                                              MAINS137
C
C CALL THE AIRCRAFT SOURCE ROUTINES AND PRINT THE          MAINS138
C WIND DEPENDENT INPUT                                     MAINS139
C                                                       MAINS140
C
C CALL ACSRCE                                              MAINS141
C CALL DEPINP                                              MAINS142
C IWDT = IWDT                                              MAINS143
C
C IF STATISICAL OPTION IS CHOSEN, WRITE FIRST RECORD ON NSTAPE MAINS144
C
C IF (NRSTAT.LE.0) GO TO 300                                MAINS145
C NHR=1                                                       MAINS146
C WRITE (NSTAPE) TITLE1,IHR1,NRSTAT,((RSTAT(I,J),I=1,2),J=1,NRSTAT), MAINS147
C . IWD,WD,IWS,WS,JSTAB,HLID,NHR
C 300 CONTINUE                                              MAINS148
C
C CALL THE DIFFUSION MODEL                                 MAINS149
C
C CALL FCISOR                                              MAINS150
C IF (NFSTAT.EQ.0) GO TO 320                                MAINS151
C DC 310 K=1,3                                              MAINS152
C PUNCH 301,NRECEP,K                                         MAINS153
C 301 FORMAT(2I6)                                           MAINS154
C DC 310 N=1,NRECEP                                         MAINS155
C PUNCH 302,(RECEP(I,N),I=1,2),(RECDAT(K,J,N),J=1,6)      MAINS156
C 302 FORMAT(1P8E10.3)                                     MAINS157
C 310 CCNTINUE                                              MAINS158
C 320 CONTINUE                                              MAINS159
C
C PRINT RESULTS                                             MAINS160
C
C CALL CPUTPUT                                              MAINS161
C
C IF THE STATISTICAL OPTION IS CHOSEN, RECORD THE OUTPUT MAINS162
C
C IF (NRSTAT.EQ.0) GO TO 360                                MAINS163
C K=4                                                       MAINS164
C PUNCH 301,NRECEP,K                                         MAINS165
C DO 330 N=1,NRECEP                                         MAINS166
C PUNCH 302,(RECEP(I,N),I=1,2),(RECDAT(1,J,N),J=1,6)      MAINS167
C 330 CCNTINUE                                              MAINS168
C IF (NRSTAT.LT.0) GO TO 360                                MAINS169
C K=0                                                       MAINS170
C DC 350 N=1,NRECEP                                         MAINS171
C IF (IRSTAT(N).LE.0) GO TO 350                           MAINS172
C K=K+1                                                       MAINS173
C DO 340 J=1,6                                              MAINS174
C RECDAT(1,J,K)=RECDAT(1,J,N)                           MAINS175
C

```

```
340 CCNTINUE  
350 CCNTINUE  
    WRITE (NSTAPE) ((REC DAT(1,J,N), J=1,6), N=1, NRSTAT)  
360 CCNTINUE  
30  CCNTINUE  
50  CCNTINUE  
    RETURN  
    END
```

```
MAINS186  
MAINS187  
MAINS188  
MAINS189  
MAINS190  
MAINS191  
MAINS192  
MAINS193
```

## SUBROUTINE METHA

### Purpose:

To calculate diurnal emissions allowing each source in a class to have a unique or default distribution pattern.

### Input:

1. The ICLASS number of the sources and NPTC, the number of sources not using the default of a uniform distribution.
2. For each of the NPTC sources, the source ID number and fractions of the hour, day and month, FH, FD and FM, which that source is on. If one or two of the fractions are left blank, the default is used. If all are blank, the source is assumed to be off.

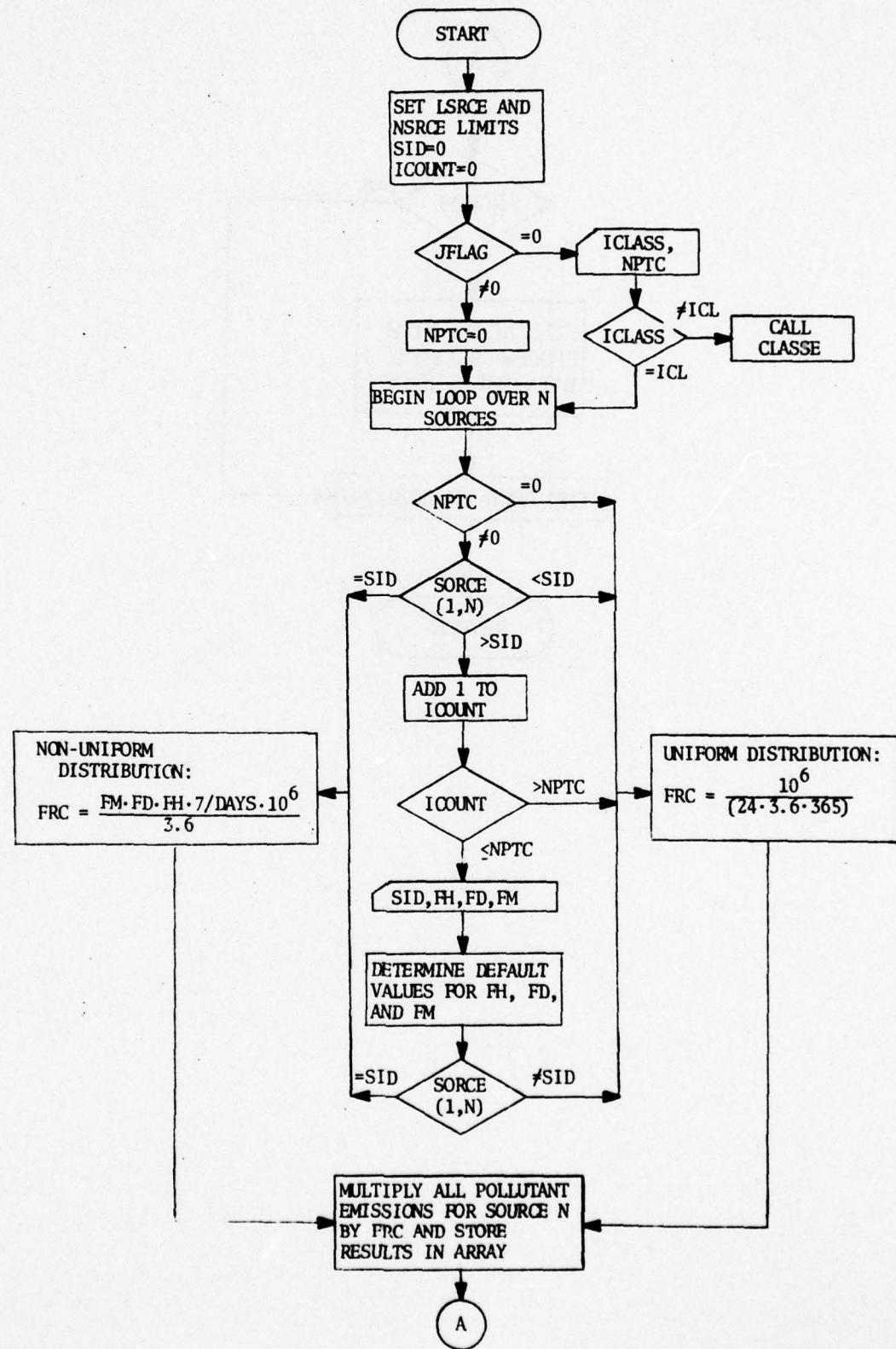
### Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

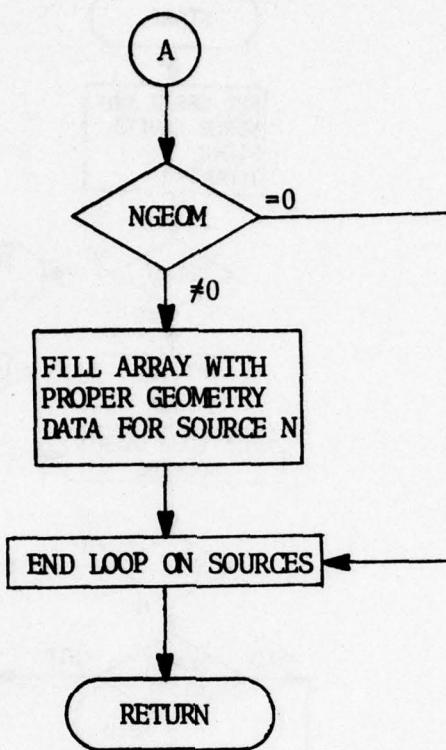
### Subroutines Called:

CLASSE

SUBROUTINE METHA



SUBROUTINE METHA (Contd.)



```

SUBROUTINE METHA(MAXN,ARRAY,I1,I2,ICL) METHA000
C THIS ROUTINE CALCULATES DIURNAL EMISSIONS ALLOWING EACH METHA001
C SOURCE IN A CLASS TO HAVE A DIFFERENT DISTRIBUTION PATTERN. METHA002
C DEFAULTS ARE: FH = 1/24 METHA003
C FD = 1/7 METHA004
C FM = 1/12 OR 1 METHA005
C
C COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG METHA006
C CCMCN /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN, METHA007
C . NACPT,NACAR,NLN,ENPT(16,100),ENAR(11,100),ENLN(14,20), METHA008
C . ABET(16,150),ABAR(11,100),ABLN(14,100) METHA009
C . COMMON/JUNK/DA SRCE,NSRCE,SORCE(17,300),SORGM(10,200) METHA010
C . ,LOC1,LOC2,NGLOC,NPT METHA011
C DIMENSION ARRAY(I1,I2) METHA012
C LSRCE=NSRCE+1 METHA013
C NSRCE=NSRCE+MAXN METHA014
C SID=0. METHA015
C ICOUNT=0 METHA016
C IF (JFLAG.EQ.0) GO TO 5 METHA017
C NPTC=0 METHA018
C GO TO 6 METHA019
C 5 REAL 1,ICLASS,NPTC METHA020
C 1 FCRMAT(2I4) METHA021
C IF (ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS) METHA022
C 6 DC 100 N=LSRCE,NSRCE METHA023
C IF (NPTC.EQ.0) GO TO 30 METHA024
C IF (SID-SORCE(1,N)) 10,40,30 METHA025
C 10 ICOUNT=ICOUNT+1 METHA026
C IF (ICOUNT.GT.NPTC) GO TO 30 METHA027
C REAL 2,SID,FH,FD,FM METHA028
C 2 FCRMAT(F4.0,4X,3F8.7) METHA029
C IF (FH+FD+FM.EQ.0.0) GO TO 20 METHA030
C
C DETERMINE DEFAULT VALUES METHA031
C
C IF (FM.NE.0.0) GO TO 15 METHA032
C FM=1./12. METHA033
C IF (DAYS.GE.365.) FM=1. METHA034
C 15 IF (FD.EQ.0.0) FD=1./7. METHA035
C IF (FH.EQ.0.0) FH=1./24. METHA036
C
C 20 CONTINUE METHA037
C IF (SID-SORCE(1,N)) 30,40,30 METHA038
C
C UNIFORM DISTRIBUTION METHA039
C
C 30 FRC=1.0E+6/(24.*3.6*365.) METHA040
C GO TO 50 METHA041
C
C NCN-UNIFORM DISTRIBUTION METHA042
C
C 40 FRC=FM*FD*FH*(7./DAYS)*(1.0E+6/3.6) METHA043
C 50 DC 60 I=1,NPLTS METHA044
C ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC METHA045
C 60 CCNTINUE METHA046
C IF (NGECM.EQ.0) GO TO 100 METHA047
C DC 70 I=1,NGFOM METHA048
C ARRAY(I,N)=SORCE(I+2,N) METHA049
C 70 CCNTINUE METHA050
C IF (IFT.EQ.1) ARRAY(10,N)=SORCE(2,N) METHA051
C 100 CCNTINUE METHA052

```

RETURN  
END

METHA062  
METHA063

## SUBROUTINE METHB

### Purpose:

To calculate diurnal emissions using a degree-hour method.

### Input:

The ICLASS number of the sources and UNIFRC, the fraction of emissions which are to be uniformly distributed.

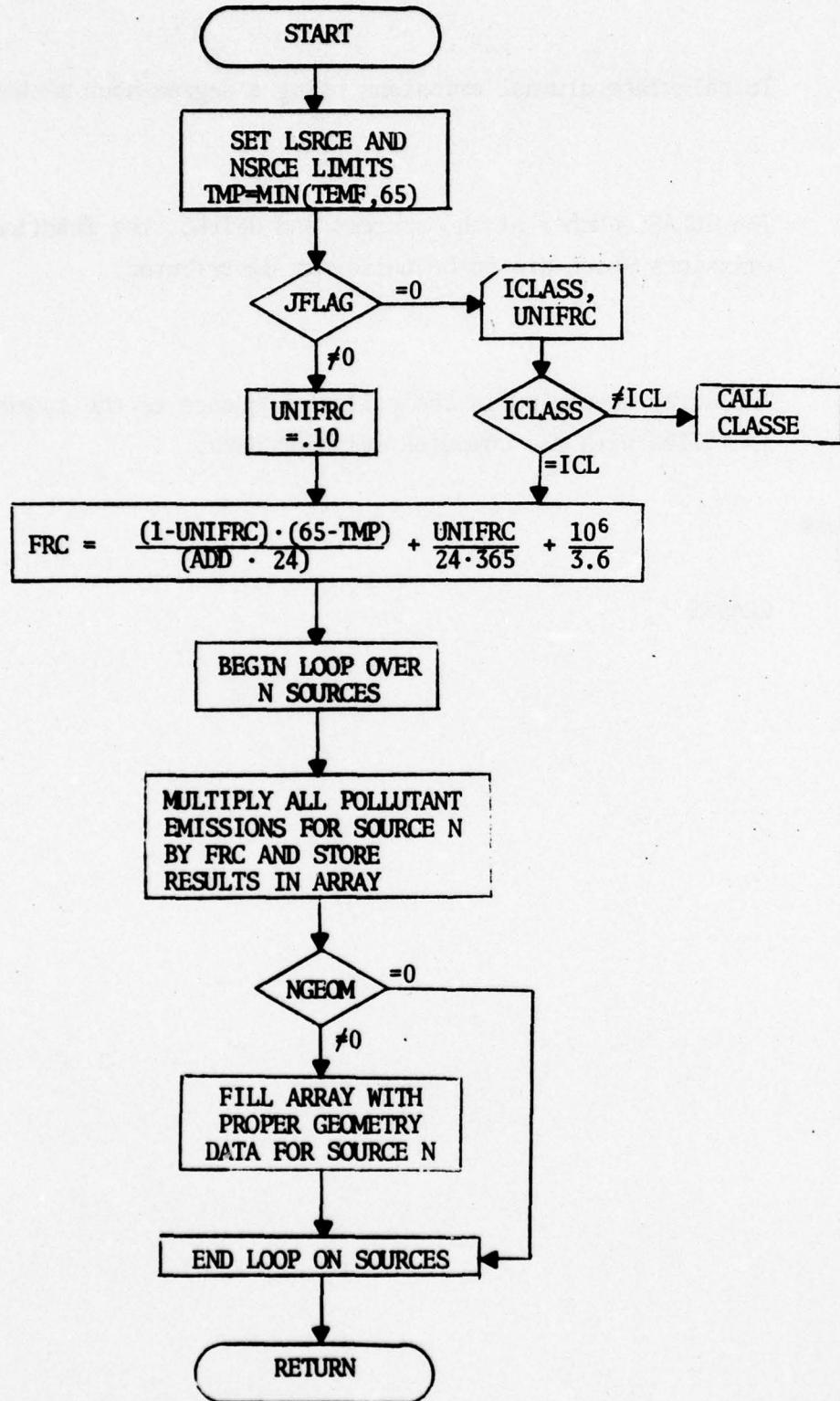
### Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

### Subroutines Called:

CLASSE

SUBROUTINE METHB



```

C SUBROUTINE METHB(MAXN,ARRAY,I1,I2,ICL) METHB000
C THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING METHB001
C A DEGREE-HOUR METHOD METHB002
C
C COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG METHB003
C COMMON /SPCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN, METHB004
C . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20), METHB005
C . ABPT(16,150),ABAR(11,100),ABLN(14,100) METHB006
C COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200) METHB007
C . LOC1,LOC2,NGEOM,IPT METHB008
C COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD, METHB009
C . JSTAEC,HLID,TEMF,TEMK METHB010
C DIMENSION ARRAY(I1,I2) METHB011
C COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR METHB012
C LSRCE=NSRCE+1 METHB013
C NSRCE=NSRCE+MAXN METHB014
C TMP=TEMF METHB015
C IF (TEMF.GT.65.) TMP=65. METHB016
C IF (JFLAG.EQ.0) GO TO 5 METHB017
C UNIFRC=.10 METHB018
C GO TO 6 METHB019
5 READ 1,ICLASS,UNIFRC METHB020
1 FFORMAT(I4,4X,F8.7) METHB021
1 IF(ICLASS.NE.ICL) CALL CLASSE(ICL,ICLASS) METHB022
6 FRC=(((1.0-UNIFRC)*((65.0-TMP)/(ADD*24.0)))+(UNIFRC/(24.0*365.0))) METHB023
. * (1.0E+6/3.6) METHB024
C
C DC 30 N=LSRCE,NSRCE METHB025
C DC 10 I=1,NPLTS METHB026
C ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC METHB027
10 CONTINUE METHB028
C IF (NGEOM.EQ.0) GO TO 30 METHB029
C DC 20 I=1,NGEOM METHB030
C ARRAY(I,N)=SORCE(I+2,N) METHB031
20 CCNTINUE METHB032
30 CCNTINUE METHB033
C RETURN METHB034
C END METHB035
METHB036
METHB037
METHB038

```

## SUBROUTINE METHC

### Purpose:

To calculate diurnal emissions using the same distribution pattern for all sources in the class.

### Input:

The ICLASS number of the sources and the fractions of the hour, day and month, FH, FD and FM, which the sources are on. If one or two of the fractions are left blank, the default is used. If all are blank, the sources are assumed to be off.

### Output:

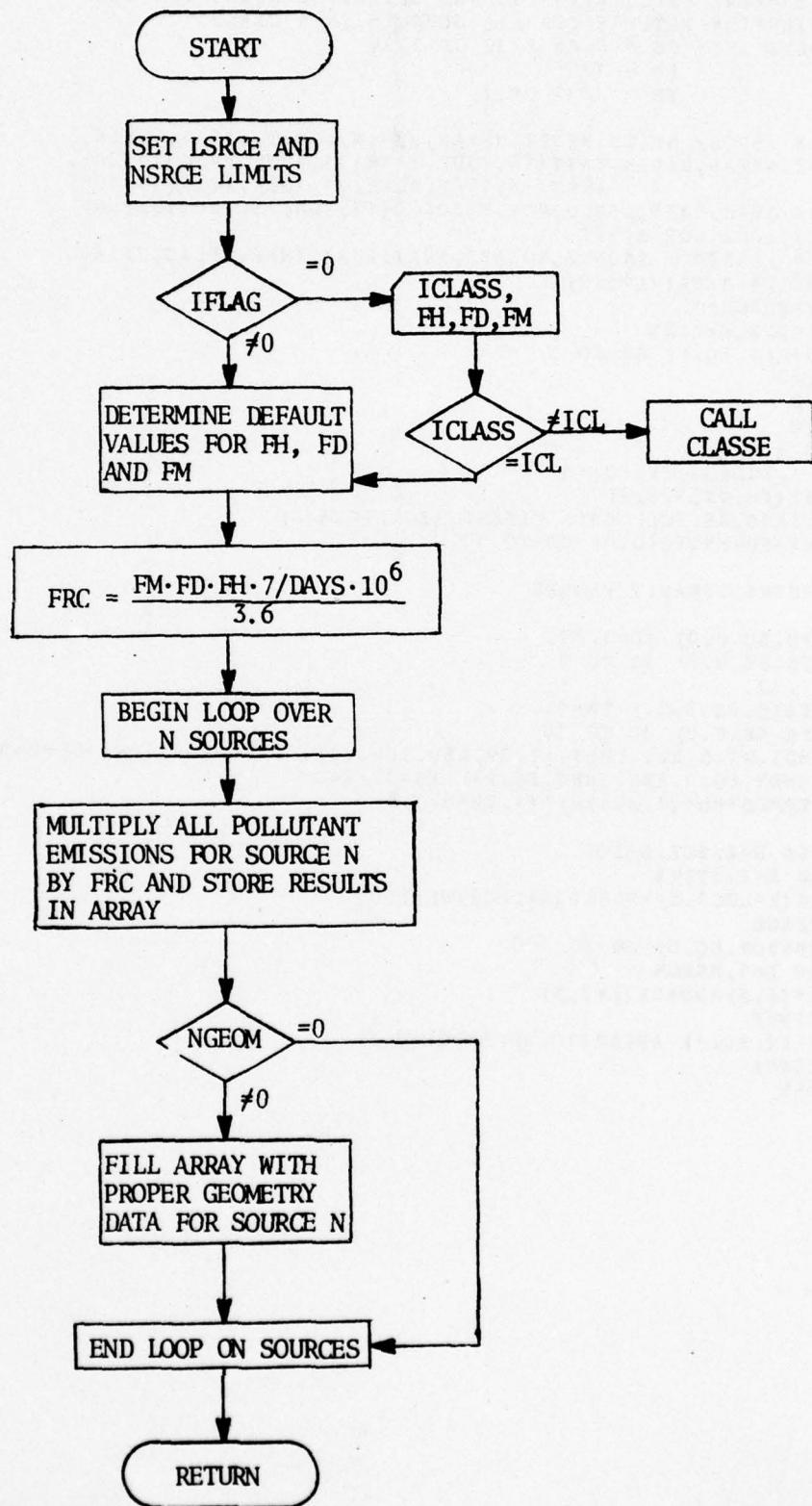
The array specified in the calling sequence to the subroutine is filled with the computed emission data.

### Subroutines

#### Called:

CLASSE

SUBROUTINE METHC



```

C SUBROUTINE METHC(MAXN,ARRAY,I1,I2,ICL)          METHC000
C THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE SAME      METHC001
C DISTRIBUTION PATTERN FOR ALL SOURCES IN A CLASS.          METHC002
C DEFAULTS ARE: FH = 0 OR 1/12 OR 1/24          METHC003
C          FD = 1/7          METHC004
C          FM = 1/12 OR 1          METHC005
C          METHC006
C          METHC007
C          METHC008
C          METHC009
C          METHC010
C          METHC011
C          METHC012
C          METHC013
C          METHC014
C          METHC015
C          METHC016
C          METHC017
C          METHC018
C          METHC019
C          METHC020
C          METHC021
C          METHC022
C          METHC023
C          METHC024
C          METHC025
C          METHC026
C          METHC027
C          METHC028
C          METHC029
C          METHC030
C          METHC031
C          METHC032
C          METHC033
C          METHC034
C          METHC035
C          METHC036
C          METHC037
C          METHC038
C          METHC039
C          METHC040
C          METHC041
C          METHC042
C          METHC043
C          METHC044
C          METHC045
C          METHC046
C          METHC047
C          METHC048
C          METHC049

COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
. NACPT,NACAR,NACIN,ENET(16,100),ENAR(11,100),ENLN(14,20),
. ABPT(16,150),ABAR(11,100),ABLN(14,100)
. CCMC/NJUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
. ,LOC1,LOC2,NGECM,IPT
COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
DIMENSION ARRAY(I1,I2)
LSRCE=NSRCE+1
NSRCE=NSRCE+MAXN
IF (JFLAG.EQ.0) GO TO 5
FD=1./7.
FM=0.0
FH=0.0
GC TC 6
5 READ 1,ICLASS,FH,FD,FM
1 FORMAT(I4,4X,3F8.7)
IF (ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS)
IF (FH+FD+FM.EQ.0.0) GO TO 10
C DETERMINE DEFAULT VALUES
C
IF (FD.EQ.0.0) FD=1./7.
6 IF (FM.NE.0.0) GO TO 7
FM=1./12.
IF (DAYS.GE.365.) FM=1.
7 IF (FH.NE.0.0) GO TO 10
IF (IHR1.GT.6.AND.IHR1.LT.19.AND.IHR2.GT.6.AND.IHR2.LT.19) FH=1./12.
IF (IHR1.EQ.1.AND.IHR2.EQ.24) FH=1./24.
10 FRC=FM*FD*FH*(7./DAYS)*(1.0E+6/3.6)
C
20 DC 100 N=LSRCE,NSRCE
DO 30 I=1,NFLTS
ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC
30 CCNTINUE
IF (NGEOM.EQ.0) GO TO 100
DO 40 I=1,NGEOM
ARRAY(I,N)=SORCE(I+2,N)
40 CCNTINUE
IF (I.EQ.1) ARRAY(10,N)=SORCE(2,N)
100 CONTINUE
RETURN
END

```

## SUBROUTINE METHOD

### Purpose:

To calculate diurnal emissions using the temporal distribution arrays for fuel handling activities.

### Input:

None

### Output:

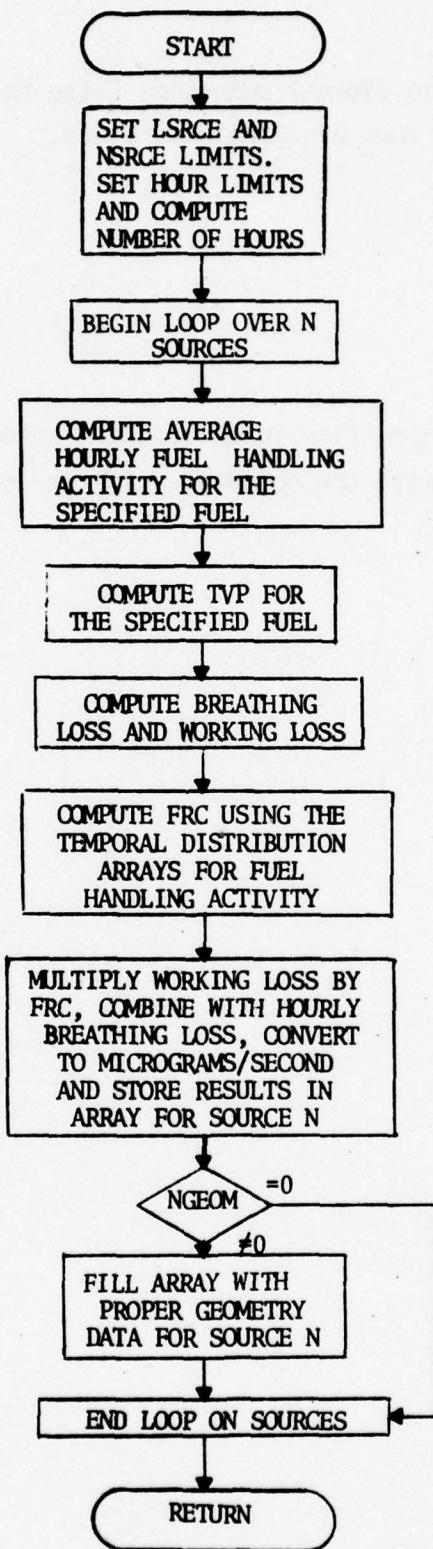
The array specified in the calling sequence to the subroutine is filled with the computed emission data.

### Subroutines

#### Called:

None

SUBROUTINE METHD



```

      SUBROUTINE METHD(MAXN,ARRAY,I1,I2)                               METHD000
C
C      THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE          METHD001
C      TEMPORAL DISTRIBUTION ARRAYS FOR FUEL HANDLING ACTIVITIES      METHD002
C
C      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,          METHD003
C      . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),      METHD004
C      . ABPT(16,150),ABAR(11,100),ABLN(14,100)                      METHD005
C      COMMON/JUNK/DAYS,LSRCE,NSPCE,SORCE(17,300),SORGM(10,200)        METHD006
C      . ,LOC1,LOC2,NGECM,IFT                                         METHD007
C      COMMON/EEPIOD/IMONTH,NODAYS,IDAY,IHR1,IHR2                      METHD008
C      COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD,                      METHD009
C      . JSTAEC,HLID,TEMF,TEMK                                         METHD010
C      COMMON / DEFLAT / ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7) METHD011
C      COMMON/DSTRET/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),      METHD012
C      . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13), METHD013
C      . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1      METHD014
C      COMMON/MONMET/ TMBAR                                         METHD015
C      DIMENSION ARRAY(I1,I2)                                         METHD016
C      LSRCE=NSRCE+1                                         METHD017
C      NSRCE=NSRCE+MAXN                                         METHD018
C      NHI=IHR2                                         METHD019
C      IF(IHR1.GT.IHR2) NHI=24+IHR2                         METHD020
C      HRS=NHI-IHR1 + 1                                         METHD021
C      DC 30 N=LSRCE,NSRCE                                         METHD022
C      FLHOUR=0.                                         METHD023
C      IDF=SCRCE(14,N)                                         METHD024
C      DC 10 I=IHR1,NHI                                         METHD025
C      II=I                                         METHD026
C      IF(I.GT.24) II=I-24                                         METHD027
C      FLHOUR=FLHR(II,IDF)+FLHOUR                         METHD028
C      10 CCNTINUE                                         METHD029
C      FLHCUR=FLHCUR/HRS                                         METHD030
C      TVP=EXP(ALPHA(IDF)-BETA(IDF)/(5.* (TMBAR-32.)/9.+273.))      METHD031
C      BRLCSS=SOFCE(13,N)*(TVP/(14.7-TVP))**0.69          METHD032
C      WRKLOS=SOFCE(12,N)*TVE                                METHD033
C      FRC=FLMO(IMONTH,IDF)*FLDY(IDAY,IDF)*FLHOUR*(7./DAYS)      METHD034
C
C      ARRAY(12,N)=(BRLOSS/(365.*24.)+WRKLOS*FRC)*1.E+6/3.6      METHD035
C      IF(IPT.EQ.1) ARRAY(10,N)=SOFCE(2,N)                      METHD036
C      IF (NGEOM.EQ.0) GO TO 30                                METHD037
C      DC 20 I=1,NGEOM                                         METHD038
C      ARRAY(I,N)=SOFCE(I+2,N)                                METHD039
C      20 CCNTINUE                                         METHD040
C      30 CCNTINUE                                         METHD041
C      RETURN                                         METHD042
C      END                                         METHD043

```

SUBROUTINE METHE

Purpose:

To calculate diurnal emissions using the temporal distribution arrays for vehicle activities.

Input:

None

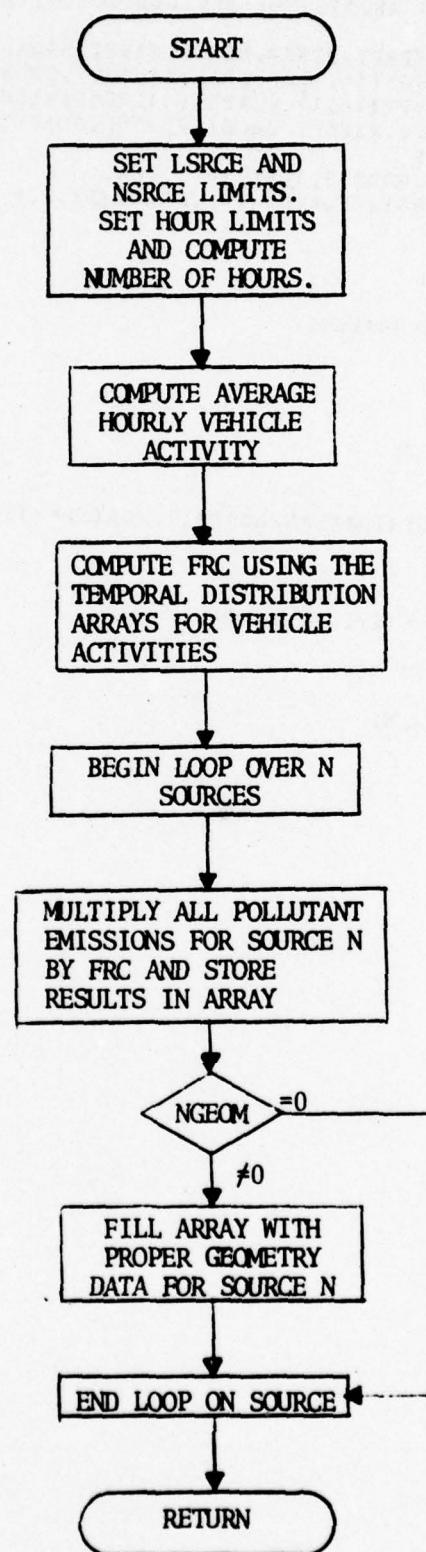
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

Subroutines  
Called:

None

SUBROUTINE METHE



```

C SUBROUTINE METHE(MAXN,ARRAY,ARMO,ARDY,ARHR,I1,I2) METHE000
C THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE METHE001
C TEMPORAL DISTRIBUTION ARRAYS FOR VEHICLE ACTIVITES METHE002
C METHE003
C METHE004
COMMON /SRCE/ NPLTS,NENET,NENAR,NENLN,NABPT,NABAR,NABLN, METHE005
NACPT,NACAR,NACLN,ENFT(16,100),ENAR(11,100),ENLN(14,20), METHE006
ABPT(16,150),ABAR(11,100),ABLN(14,100) METHE007
COMMON/JUNK/DAYS,LSRCE,NSRCF,SORCE(17,300),SORGM(10,200) METHE008
,LOC1,LOC2,NGECM,IPT METHE009
CMMCN/PERIOD/IMONTH,NODAYS,IDAY,IHR1,IHR2 METHE010
DIMENSION ARMO(13),ARDY(2),ARHR(24),ARRAY(11,12) METHE011
LSRCE=NSRCE+1 METHE012
NSRCE=NSRCE+MAXN METHE013
ARHOUR=0. METHE014
NHI=IHR2 METHE015
IF(IHR1.GT.IHR2) NHI=24+IHR2 METHE016
HRS=NHI-IHR1+1 METHE017
DO 10 I=IHR1,NHI METHE018
II=I METHE019
IF(I.GT.24) II=I-24 METHE020
ARHOUR=ARHOUR+ARHR(II) METHE021
10 CCNTINUE METHE022
ARHOUR=ARHOUR/HRS METHE023
FRC=ARMO(IMONTH)*ARDY(IDAY)*ARHOUR*(7./DAYS)*(1E+6/3.6) METHE024
C
DO 40 N=LSRCE,NSRCE METHE025
DO 20 I=1,NPLTS METHE026
ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC METHE027
20 CCNTINUE METHE028
IF (NGEOM.EQ.C) GO TO 40 METHE029
DC 30 I=1,NGEOM METHE030
30 ARRAY(I,N)=SORCE(I+2,N) METHE031
40 CCNTINUE METHE032
RETURN METHE033
END METHE034
METHE035

```

## SUBROUTINE OUTPUT

### Purpose:

To print the pollutant concentrations at all receptors for the environ, airbase, aircraft and total combined sources.

### Input:

1. Title information.
2. The RECEP and RECDAT arrays containing receptor and concentration data.

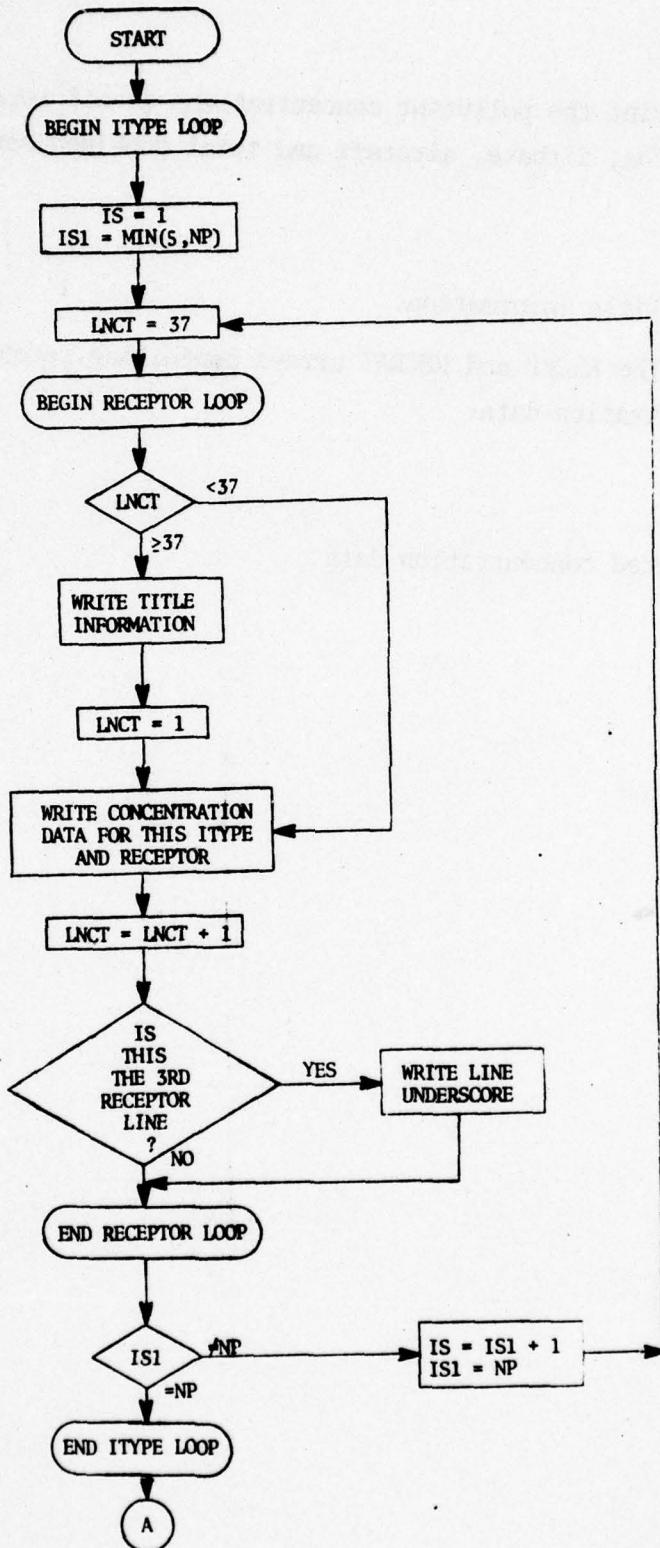
### Output:

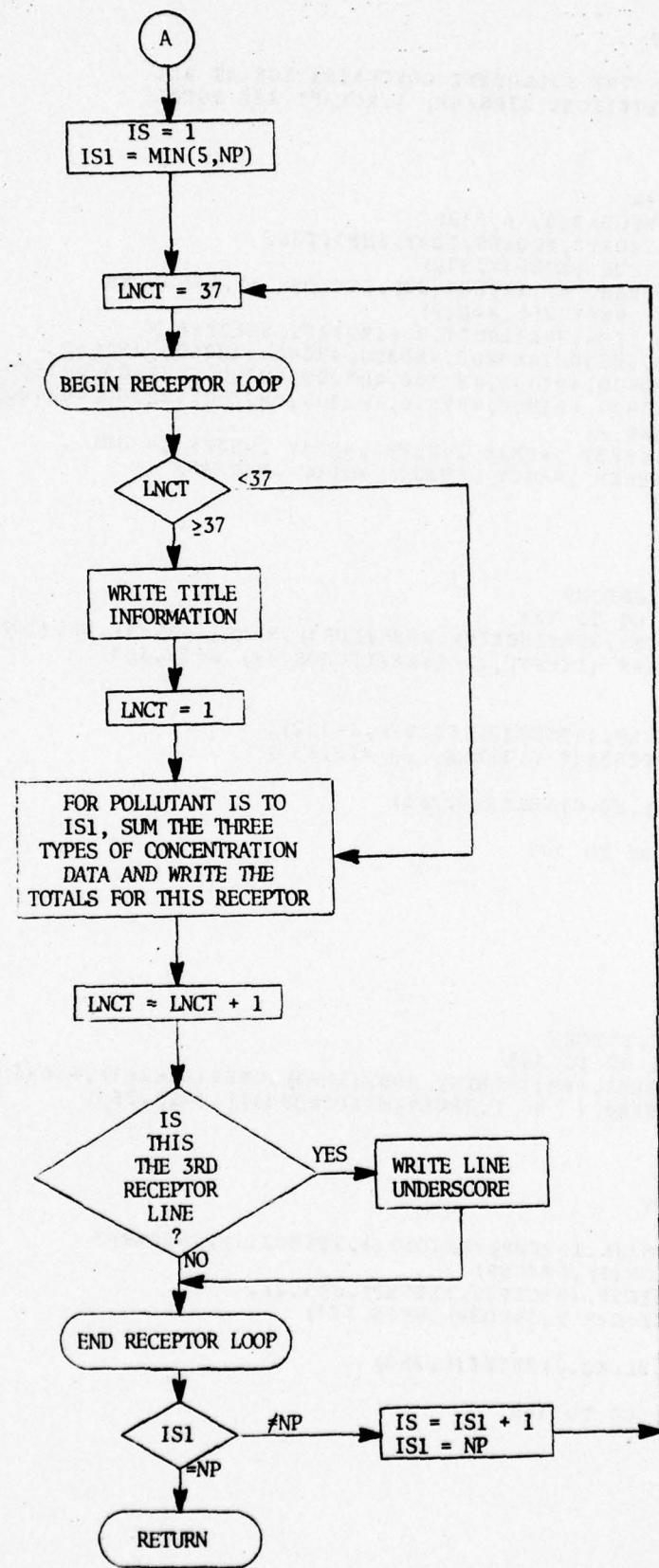
Printed concentration data.

### Subroutines Called:

None

SUBROUTINE OUTPUT





SUBROUTINE OUTPUT

C THIS ROUTINE PRINTS THE POLLUTANT CONCENTRATION AT ALL  
 C RECEPTORS FOR THE ENVIRON, AIRBASE, AIRCRAFT AND TOTAL  
 C COMBINED SOURCES.

REAL\*8 POLNAM  
 REAL\*8 SORNAME(4)  
 COMMON /AIRQAL/ RECDAT(3, 6, 312)  
 COMMON /PERIOD/ IMONTH, NODAYS, IDAY, IHR1, IHR2  
 COMMON /RCPT/ NRECEP, RECEP(2, 312)  
 COMMON /TITLE/ POLNAM( 6 ), TITLE1(20), IPCHOS( 6 ), NXPOL, NP  
 DIMENSION NNM(13), NNHR(25), NND(2)  
 DATA SORNAME/7HENIRON, 7HAIRPORT, 8HAIRCRAFT, 5HTOTAL /  
 DATA NNHR/4H0000, 4H0100, 4H0200, 4H0300, 4H0400, 4H0500, 4H0600,  
 . 4H0700, 4H0800, 4H0900, 4H1000, 4H1100, 4H1200, 4H1300, 4H1400, 4H1500,  
 . 4H1600, 4H1700, 4H1800, 4H1900, 4H2000, 4H2100, 4H2200, 4H2300, 4H2400 /,  
 . NND /4HDAY, 4HEND /,  
 . NNM/4HJAN, 4HFEB, 4HMAR, 4HAPR, 4HMAY, 4HJUN, 4HJUL,  
 . 4HAUG, 4HSEP, 4HOCT, 4HNOV, 4HDEC, 4HYEAR/  
 DO 100 ITYPE=1, 3  
 IS=1  
 IS1=MIN0(5, NP)  
 110 LNCT=37  
 DO 120 IRECEP=1, NRECEP  
 IF (LNCT.LT.37) GO TO 121  
 WRITE(6, 220) TITLE1, NNM(IMONTH), NNHR(IHR1), NNHR(IHR2+1), NND(IDAY)  
 WRITE(6, 200) SORNAME (ITYPE), (POLNAM(IPCHOS(J)), J=IS, IS1)  
 WRITE(6, 260)  
 LNCT=1  
 121 WRITE(6, 210) IRECEP, (RECEP(J, IRECEP), J=1, 2),  
 . (RECDAT(ITYPE, IPCHOS(K), IRECEP), K=IS, IS1)  
 LNCT=LNCT+1  
 IF (MOD(IRECEP, 3).EQ.0) WRITE(6, 260)  
 120 CCNTINUE  
 IF (IS1.EQ.NP) GO TO 100  
 IS=IS1+1  
 IS1=NP  
 GO TO 110  
 100 CONTINUE  
 IS=1  
 IS1=MIN0(5, NP)  
 125 LNCT=37  
 DO 130 IRECEP=1, NRECEP  
 IF (LNCT.LT.37) GO TO 133  
 WRITE(6, 220) TITLE1, NNM(IMONTH), NNHR(IHR1), NNHR(IHR2+1), NND(IDAY)  
 WRITE(6, 200) SORNAME ( 4 ), (POLNAM(IPCHOS(J)), J=IS, IS1)  
 WRITE(6, 260)  
 LNCT=1  
 133 CCNTINUE  
 DO 131 J=IS, IS1  
 DO 131 K=2, 3  
 131 RECDAT(1, IPCHOS(J), IRECEP)=RECDAT(1, IPCHOS(J), IRECEP) +  
 . RECDAT(K, IPCHOS(J), IRECEP)  
 WRITE(6, 210) IRECEP, (RECEP(J, IRECEP), J=1, 2),  
 . (RECDAT(1, IPCHOS(K), IRECEP), K=IS, IS1)  
 LNCT=LNCT+1  
 IF (MOD(IRECEP, 3).EQ.0) WRITE(6, 260)  
 130 CONTINUE  
 IF (IS1.EQ.NP) GO TO 140  
 IS=IS1+1  
 IS1=NP

OUTPT000  
 OUTPT001  
 OUTPT002  
 OUTPT003  
 OUTPT004  
 OUTPT005  
 OUTPT006  
 OUTPT007  
 OUTPT008  
 OUTPT009  
 OUTPT010  
 OUTPT011  
 OUTPT012  
 OUTPT013  
 OUTPT014  
 OUTPT015  
 OUTPT016  
 OUTPT017  
 OUTPT018  
 OUTPT019  
 OUTPT020  
 OUTPT021  
 OUTPT022  
 OUTPT023  
 OUTPT024  
 OUTPT025  
 OUTPT026  
 OUTPT027  
 OUTPT028  
 OUTPT029  
 OUTPT030  
 OUTPT031  
 OUTPT032  
 OUTPT033  
 OUTPT034  
 OUTPT035  
 OUTPT036  
 OUTPT037  
 OUTPT038  
 OUTPT039  
 OUTPT040  
 OUTPT041  
 OUTPT042  
 OUTPT043  
 OUTPT044  
 OUTPT045  
 OUTPT046  
 OUTPT047  
 OUTPT048  
 OUTPT049  
 OUTPT050  
 OUTPT051  
 OUTPT052  
 OUTPT053  
 OUTPT054  
 OUTPT055  
 OUTPT056  
 OUTPT057  
 OUTPT058  
 OUTPT059  
 OUTPT060  
 OUTPT061

GC TO 125 OUTPT062  
140 CCNTINUE OUTPT063  
200 FORMAT(1H0,96(1H-)/2H I,22X,33HRECEPTOR CONCENTRATION DATA FROM , OUTPT064  
. A8,8H SOURCES,23X,1HI/2H I,94(1H-),1HI/ OUTPT065  
. 37H I RECEPTOR I RECEPTOR LOCATION I,17X,24HEXPECTED ARITHMETOUTPT066  
. IC MEAN,18X,1HI/13H I NUMBER I,23X,1HI,59X,1HI/ OUTPT067  
. 2H I,10(1H-),1HI,23(1H-),1HI,59(1H-),1HI/ OUTPT068  
. 2H I,10X,1HI,5X,12H(KILOMETERS),6X,1HI,18X,22H(MICROGRAMS/CU. METOUTPT069  
. ER),19X,1HI,/2H I,10X,1HI,5X,1HX,5X,1HI,5X,1HY, OUTPT070  
. 5X,4 (3HI ,A8,1X),3HI ,A8,2H I) OUTPT071  
210 FCRMAT(2H I,I6,4X,2(1HI,F9.3,2X),1HI,5(1PE10.3,2H I)) OUTPT072  
220 FCRMAT(1H1,9X,20A4/10H MONTH = ,A4,12H PERIOD = ,A4,4H TO , OUTPT073  
. A4,16H HOURS ON A WEEK,A4) OUTPT074  
260 FORMAT(2HI,10(1H-),1HI,7(11(1H-),1HI)) OUTPT075  
RETURN OUTPT076  
END OUTPT077

## SUBROUTINE PLRISE

### Purpose:

To calculate the effective height and the vertical and horizontal dispersion coefficients for a given stack.

### Input:

The stack parameters and current meteorological conditions.

### Output:

1. The effective height,  $h_{eff}$ .
2. The vertical and horizontal dispersion coefficients,  $\sigma_{yo}$  and  $\sigma_{zo}$ .
3. KSTAB, a flag used in the TRAN function
  - = 0, the modified stack height is below the lid
  - = 1, the modified stack height is initially above the lid
  - = 2, the plume will penetrate the lid.

### Procedure:

1. For point sources having no plume rise:

$$h_{eff} = \max (Z_S, H_B, \Delta Z/2.)$$

$$\sigma_{yo} = \Delta Y/2.4$$

$$\sigma_{zo} = \Delta Z/2.4$$

$$KSTAB = 0 \text{ or } 1$$

2. For point sources which may undergo plume rise:

- a. Estimate the wind speed at the top of the aerovane
- b. Modify the stack height by the effect of the stack downwash

- c. Test for building downwash effects. If downwash occurs:

$$h_{eff} = H_B + .5L_B$$

$$\sigma_{yo} = \sigma_{zo} = h_{eff}/1.2$$

$$KSTAB = 0 \text{ or } 1$$

- d. Test to determine if the buoyant plume rise is significant.

e. Check for an inversion

f. Compute the plume rise using function RISE

g. If no downwash occurs:

$$H_{\text{eff}} = z_S + 2 \left( \frac{V_S}{U_a} - 1.5 \right) \cdot DS + \text{plume rise}$$

$$\sigma_{y0} = \Delta Y / 2.4$$

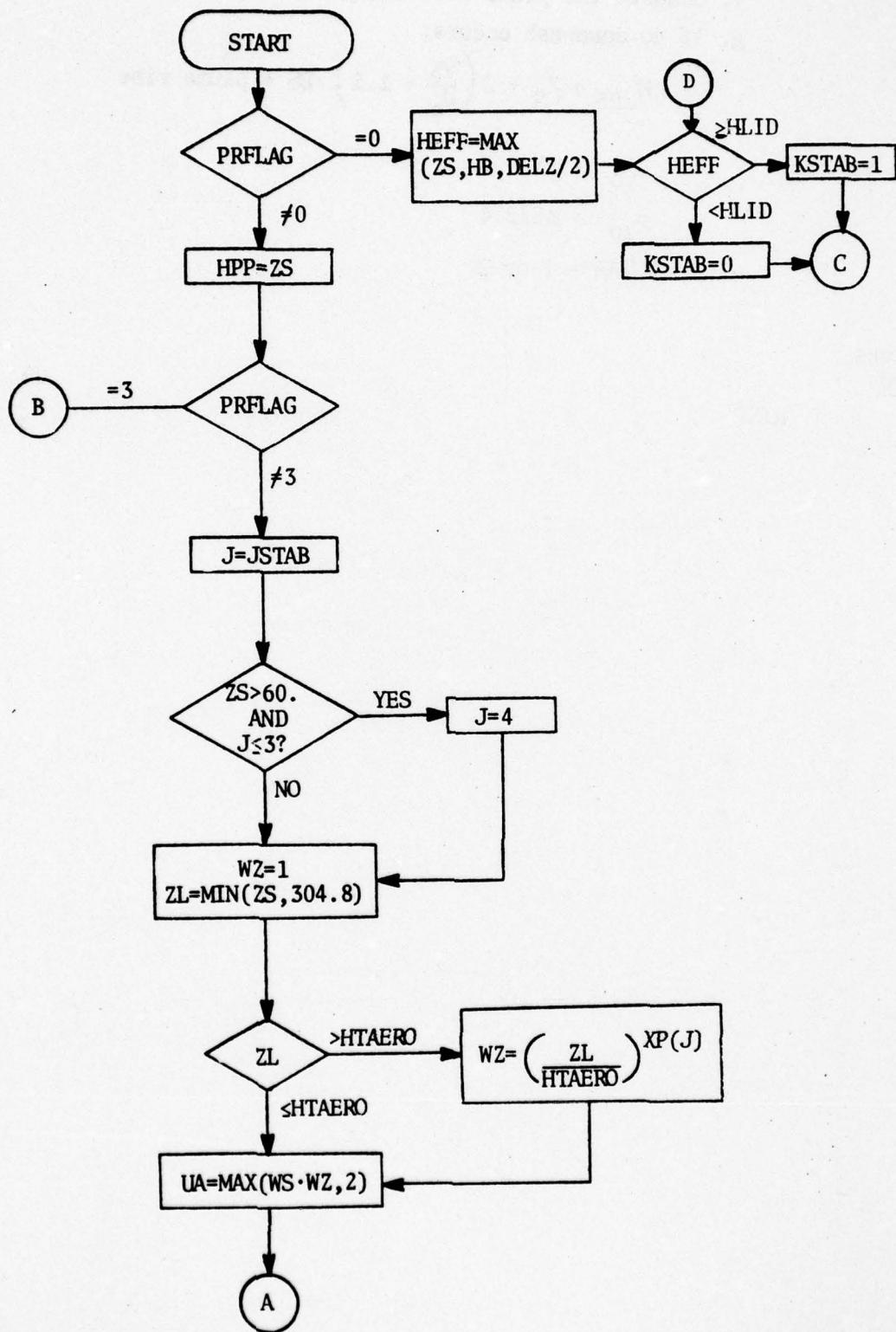
$$\sigma_{z0} = \Delta Z / 2.4$$

KSTAB = 1 or 2

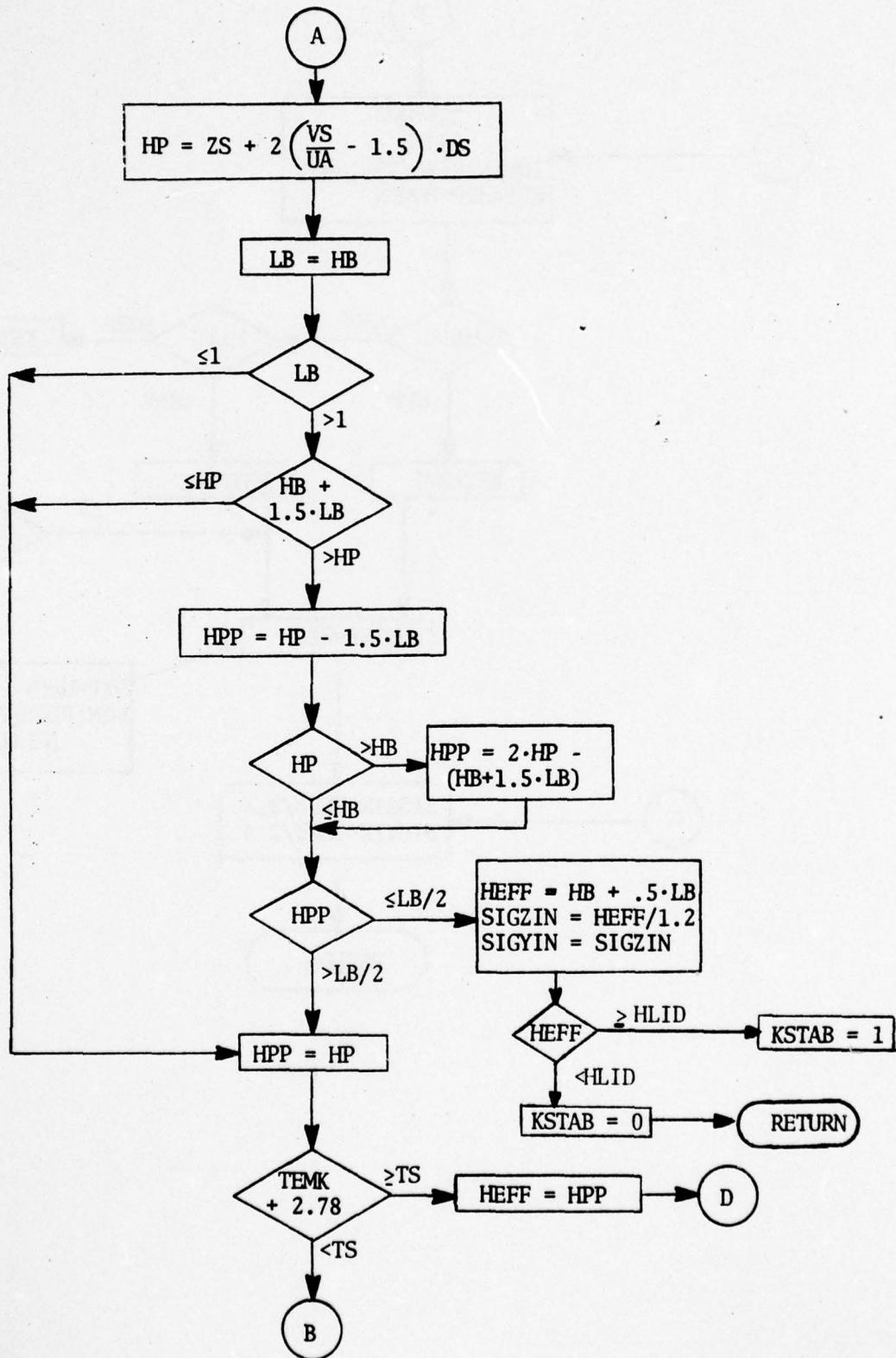
Functions  
Called:

RISE

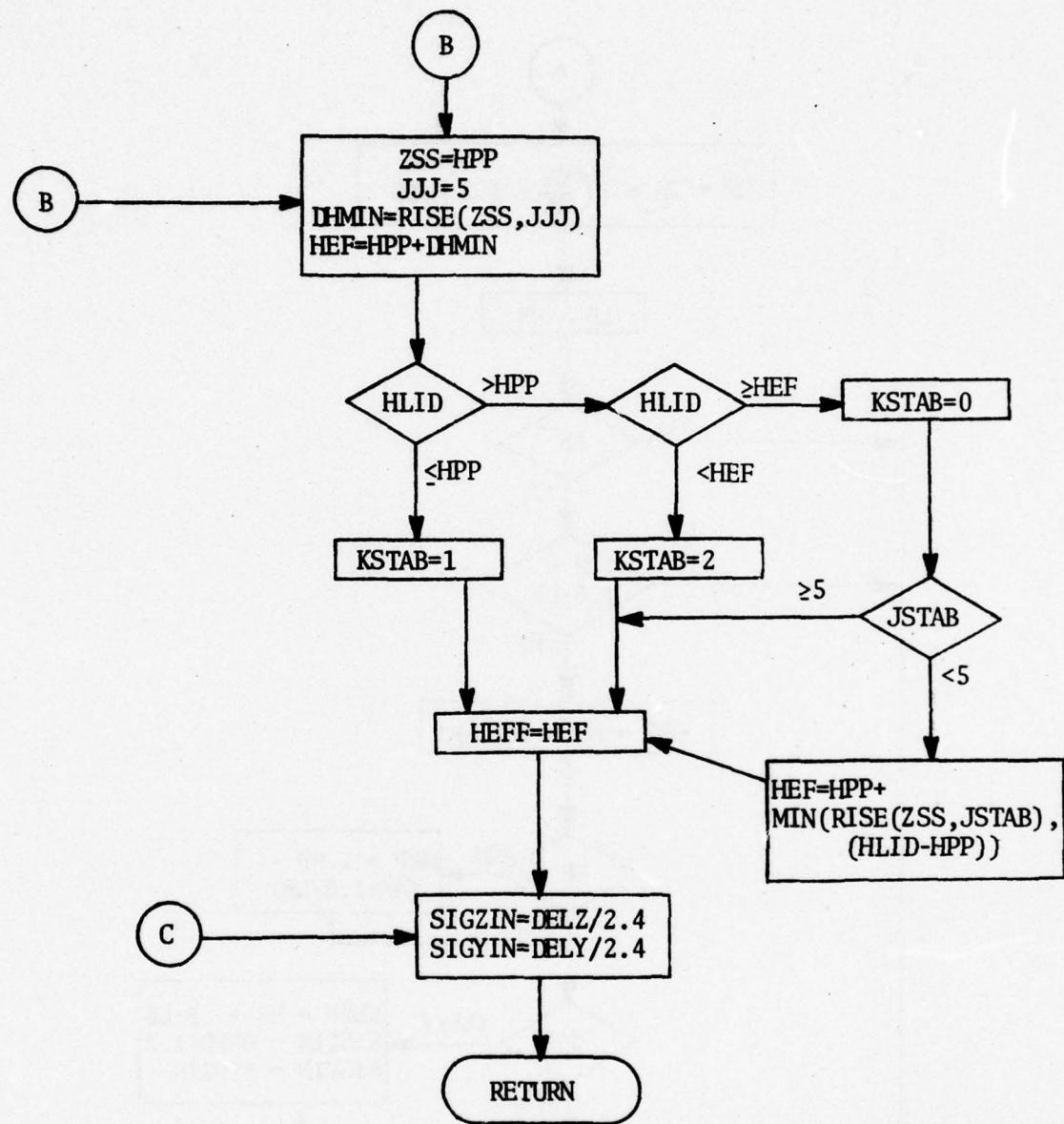
SUBROUTINE PLRISE



SUBROUTINE PLRISE (Cont'd.)



SUBROUTINE PLRISE (Cont'd.)



```

SUBROUTINE PLRSE(HEFF,KSTAR,SIGZIN,SIGYIN)          PLRSE000
C THIS SUBROUTINE CALCULATES THE FFFECTIVE HEIGHT AND THE PLRSE001
C VERTICAL AND HORIZONTAL DISPERSION COEFFICIENTS PLRSE002
C FOR A GIVEN STACK PIRSE003
C
C      REAL LP PLRSE004
C      COMMON /MET/ VS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF, PLRSE005
C      . TEMK,UA PLRSE006
C      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAFRO,XS,YS,ZS,DELY,DELZ, PLRSE007
C      . TS,VS,DS,HB,PFFLAG,EMIS(8),NPOL PLRSE008
C      COMMON /WNDPFC/ XP(6) PLRSE009
C      IF (PFFLAG.NE.0) GO TO 100 PLRSE010
C
C      FOR AN AREA SOURCE WITH A DIAMETER OF LESS THAN 50 METERS PLRSE011
C      THE EFFECTIVE EMISSION HEIGHT IS SET TO THE MAXIMUM OF PLRSE012
C      Z, THE BUILDING HEIGHT OR DELTA Z/2.0 PLRSE013
C
C      HEFF=AMAX1(ZS,HB,DELZ/2.) PLRSE014
C      50 KSTAB=0 PLRSE015
C      IF (HEFF.GE.HLID) KSTAB=1 PLRSE016
C      GO TO 230 PLRSE017
C
C      100 CONTINUE PLRSE018
C      HFP=ZS PLRSE019
C      IF (PFFLAG.EQ.3) GO TO 130 PLRSE020
C
C      FIRST TEST FOR DOWNWASH, THEN COMPUTE PLUME RISE, IF ANY PLRSE021
C
C      FOR TALL STACKS USE STABILITY 4 IN THE WIND PROFILE LAW PLRSE022
C
C      J=JSTAB PLRSE023
C      IF (ZS.GT.60..AND.J.LE.3) J=4 PLRSE024
C
C      COMPUTE THE WINDSPEED AT THE ELEVATION OF THE STACK PIRSE025
C
C      WZ=1.0 PLPSE026
C      ZL=AMIN1(ZS,304.8) PLRSE027
C      IF (ZL.GT.HTAERO) WZ=(ZL/HTAERO)**XP(J) PLRSE028
C      UA=AMAX1(VS*WZ,2.0) PLRSE029
C
C      COMPUTE STACK DOWNWASH PLRSE030
C
C      HE=2S+2.0*(VS/UA-1.5)*DS PLRSE031
C      LB=HP PLRSE032
C
C      BUILDING DOWNWASH TESTS PLPSE033
C
C      IF (LB.LE.1.) GO TO 110 PLRSE034
C      IF (HE.GE.(HB+1.5*LB)) GO TO 110 PLRSE035
C      HPP=HE-1.5*LB PLRSE036
C      IF (HP.GT.HB) HPP=2.0*HP-(HB+1.5*LB) PLRSE037
C      IF (HPP.GT.(LP/2.0)) GO TO 110 PLRSE038
C
C      COMPUTE DOWNWASH OCCURS PLRSE039
C
C      HEFF=HE+0.5*LP PLRSE040
C      SIGZIN=HEFF/1.2 PLRSE041
C      SIGYIN=SIGZIN PLRSE042
C      KSTAB=0 PLPSE043
C      IF (HEFF.GE.HLID) KSTAB=1 PLRSE044
C      RETURN PLRSE045
C
C      BUILDING DOWNWASH OCCURS PLRSE046
C
C      HEFF=HE+0.5*LP PLRSE047
C      SIGZIN=HEFF/1.2 PLRSE048
C      SIGYIN=SIGZIN PLRSE049
C      KSTAB=0 PLRSE050
C      IF (HEFF.GE.HLID) KSTAB=1 PLRSE051
C      RETURN PLRSE052
C
C      BUILDING DOWNWASH OCCURS PLPSE053
C
C      HEFF=HE+0.5*LP PLRSE054
C      SIGZIN=HEFF/1.2 PLRSE055
C      SIGYIN=SIGZIN PLRSE056
C      KSTAB=0 PLRSE057
C      IF (HEFF.GE.HLID) KSTAB=1 PLRSE058
C      RETURN PIRSE059
C
C      BUILDING DOWNWASH OCCURS PLRSE060
C
C      HEFF=HE+0.5*LP PLRSE061
C      SIGZIN=HEFF/1.2
C      SIGYIN=SIGZIN
C      KSTAB=0
C      IF (HEFF.GE.HLID) KSTAB=1
C      RETURN

```

```

C      NC BUILDING DOWNWASH, TEST FOR PLUME RISE          PIRSF062
C      110 HPP=HP                                         PLRSE063
C          IF (TS.GT.(TEMK+2.78)) GO TO 130               PLRSE064
C      COLD PLUME                                         FIRSF065
C          HEFF=HPP                                         PLPSE066
C          GC TC 50                                         PLRSE067
C          130 CONTINUE                                     PLPSE068
C          PLUME RISE EXPECTED TO BE SIGNIFICANT          PIRSE069
C          CALCULATE MINIMUM PLUME RISE                  PLRSE070
C          ZSS=HPP                                         PLRSE071
C          JJJ=5                                           PLRSE072
C          DHMIN=RISE(ZSS,JJJ)                           PLRSE073
C          HEP=HPP+DHMIN                                 PLRSE074
C          TEST FOR INTERFERENCE OF LID WITH MODIFIED PHYSICAL STACK
C          HEIGHT AND PLUME                            PLRSE075
C          IF (HLID.GT.HPP) GO TO 220                  PLRSE076
C          LID INTERFERES WITH STACK HEIGHT, USE STABILITY 5 WITH
C          INFINITE LID HEIGHT                         PIRSE077
C          KSTAB=1                                         PLRSE078
C          GO TO 225                                     PLPSE079
C          LID INTERFERES WITH PLUME, USE STABILITY 5 WITH INFINITE LID
C          220 IF (HLID.GE.HEP) GO TO 221               PIRSE080
C              KSTAB=2                                     PLRSE081
C              GO TO 225                                 PIRSE082
C          CALCULATE PLUME RISE, PLUME CANNOT PENETRATE THE LID
C          221 KSTAB=0                                     PLRSE083
C              IF (JSTAB.LT.5) HEP=HPE+AMTN1(RISE(ZSS,JSTAB),(HLID-HPP)) PIRSE084
C          225 CCNTINUE                                PLPSE085
C              HEFF=HEF                                 PLRSE086
C          230 SIGZIN=DELZ/2.4                         PIRSE087
C              SIGYIN=DELY/2.4                         PLRSE088
C              RETURN                                    PLRSE089
C              END                                     PLRSE090
C

```

## SUBROUTINE POLSOR

### Purpose:

To direct the calls to the proper diffusion routine for all input sources.

### Input:

1. Point source data for:
  - a. Environs
  - b. Airbase
  - c. Aircraft
2. Area source data for:
  - a. Environs
  - b. Airbase
  - c. Aircraft
3. Line source data for:
  - a. Environs
  - b. Airbase
  - c. Aircraft

### Output:

SORC, a vector which contains data for the current source to be transferred to the diffusion models.

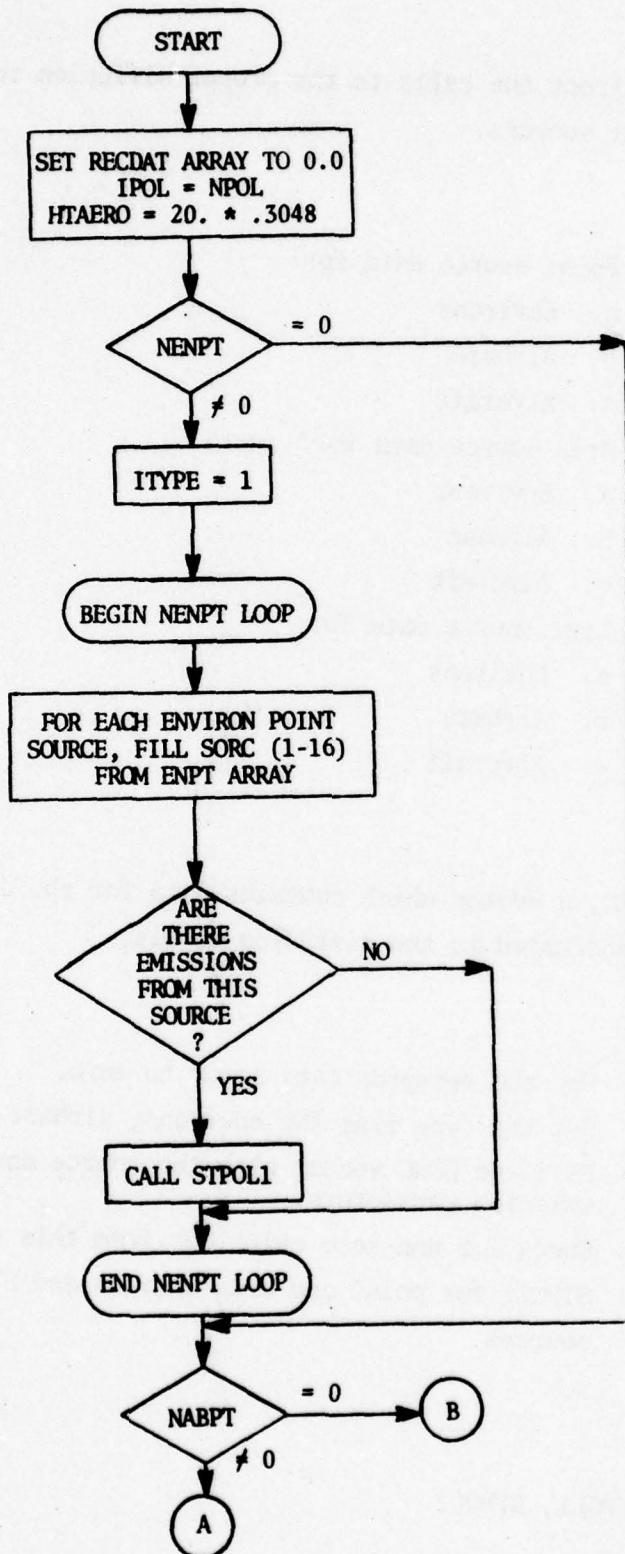
### Procedure:

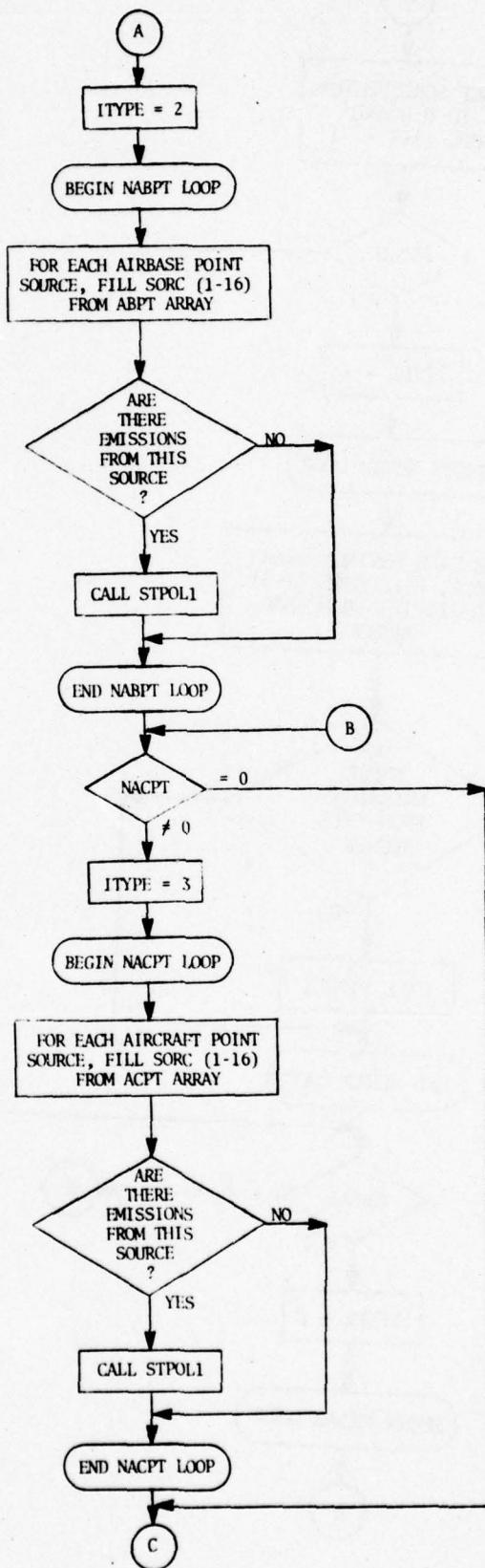
1. Set the receptor data array to zero.
2. Set the type flag for environs, airbase or aircraft.
3. Fill the SORC vector with the source description and emission parameters.
4. Check for non-zero emissions from this source and call STPOL1 for point and area sources and STPOL2 for line sources.

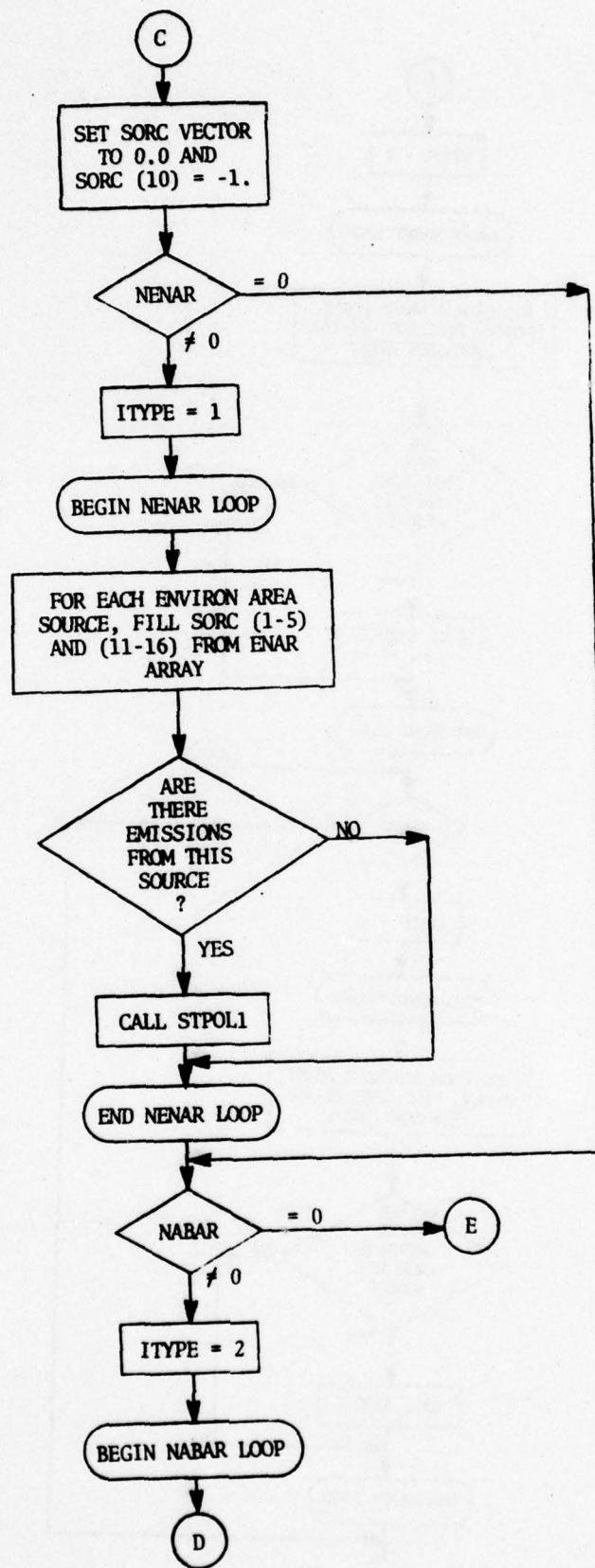
### Subroutines Called:

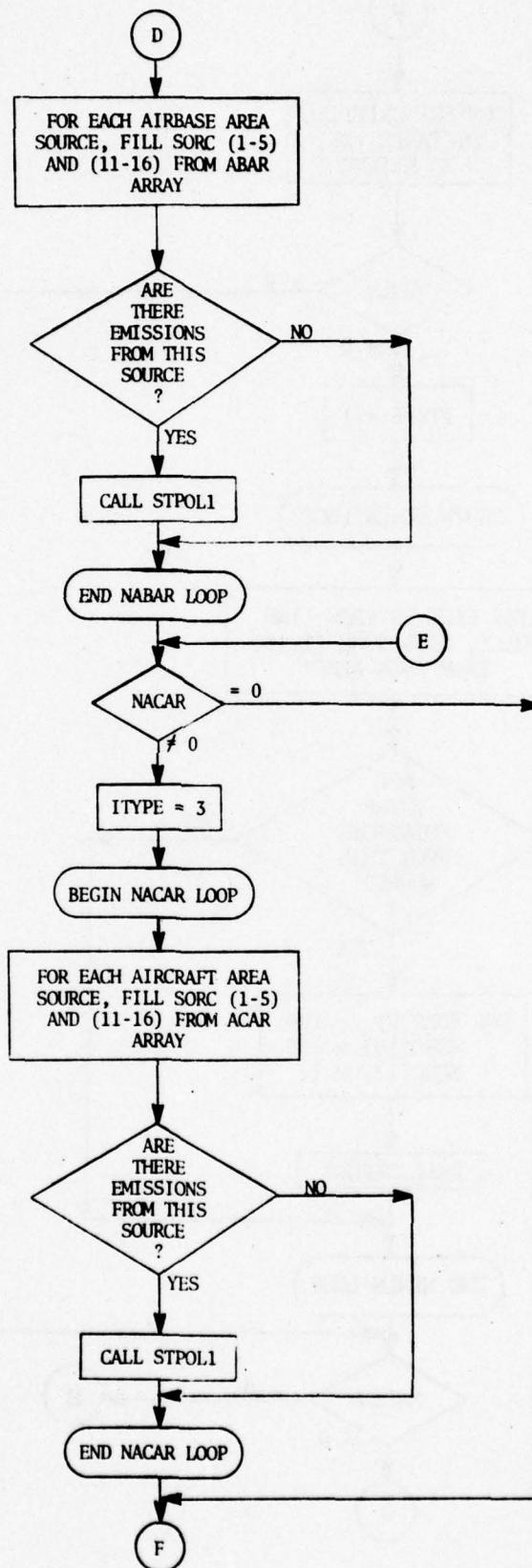
STPOL1, STPOL2

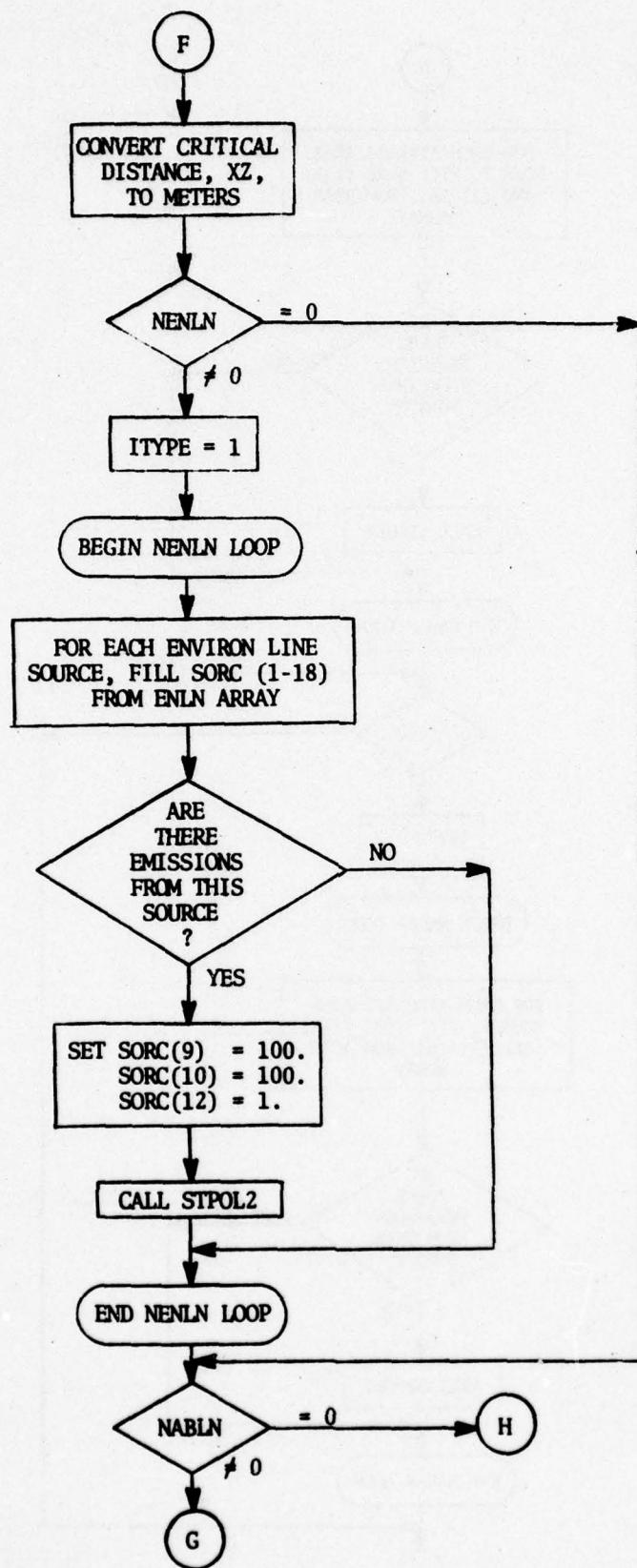
SUBROUTINE POLSOR

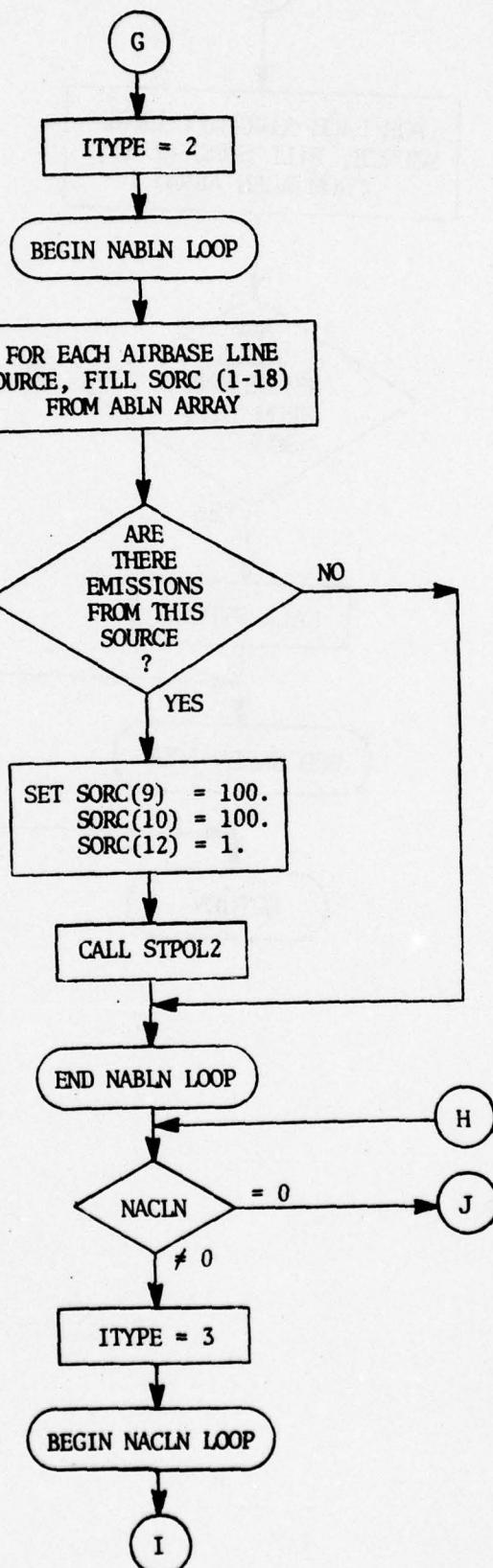


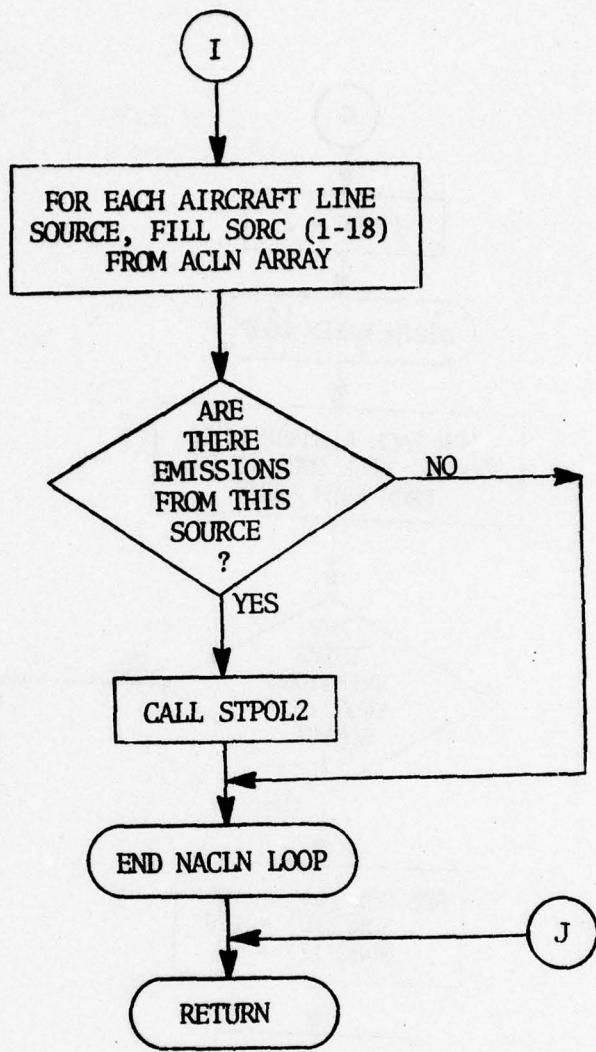












```

SUBROUTINE POLSOR
C THIS ROUTINE IS THE DRIVER FOR THE DIFFUSION MODEL. FOR ALL
C ENVIRCN, AIRBASE AND AIRCRAFT POINTS, AREAS AND LINES,
C THE SOFC VECTOR IS FILLED WITH THE APPROPRIATE SOURCE
C PARAMETERS AND THEN THE PROPER DIFFUSION ROUTINE IS CALLED
C
COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
  TEMK
COMMON /SRCE/ NPOL,NENFT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
  NACAR,NACLN,ENPT(16,100),ENAP(11,100),ENLN(14,20),ABPT(16,150),
  ABAR(11,100),ABIN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
COMMON /INFO/ IRECEP,IWDIR,ITYPE ,HTAERO,SORC(18),IPOL
COMMON /AIRQAL/ RECDAT(3, 6,312)
COMMON /XTRAN/ XZ
C
C SET RECDAT ARRAY TO 0.0
C
DO 10 I=1,3
DC 10 J=1,6
DC 10 K=1,312
10 RECDAT(I,J,K) = 0.
IFOL = NPOL
HTAERC=20.*.3048
C
C ENVIRON PCINTS
C
IF (NENFT .EQ. 0) GO TO 126
ITYPE = 1
DO 125 I=1,NENPT
DC 124 J=1,16
124 SCRC(J) = ENPT(J,I)
DO 224 J=11,16
IF (SCRC(J).NE.0.0) GO TO 225
224 CCNTINUE
GO TO 125
225 CALL STPOL1
125 CCNTINUE
C
C AIRBASE POINTS
C
126 IF (NABPT .EQ. 0) GO TO 136
ITYPE = 2
DO 135 I=1,NABPT
DC 134 J=1,16
134 SORC(J) = ABPT(J,I)
DO 234 J=11,16
IF (SCRC(J).NE.0.0) GO TO 235
234 CCNTINUE
GO TO 135
235 CALL STPOL1
135 CCNTINUE
C
C AIRCRAFT POINTS
C
136 IF (NACPT .EQ. 0) GO TO 146
ITYPE = 3
DO 145 I=1,NACPT
DO 144 J=1,16
144 SCRC(J) = ACPT(J,I)
DO 244 J=11,16
IF (SORC(J).NE.0.0) GO TO 245

```

POLSR000  
POLSR001  
POLSR002  
POLSR003  
POLSR004  
POLSR005  
POLSR006  
POLSR007  
POLSR008  
POLSR009  
POLSR010  
POLSR011  
POLSR012  
POLSR013  
POLSR014  
POLSR015  
POLSR016  
POLSR017  
POLSR018  
POLSR019  
POLSR020  
POLSR021  
POLSR022  
POLSR023  
POLSR024  
POLSR025  
POLSR026  
POLSR027  
POLSR028  
POLSR029  
POLSR030  
POLSR031  
POLSR032  
POLSR033  
POLSR034  
POLSR035  
POLSR036  
POLSR037  
POLSR038  
POLSR039  
POLSR040  
POLSR041  
POLSR042  
POLSR043  
POLSR044  
POLSR045  
POLSR046  
POLSR047  
POLSR048  
POLSR049  
POLSR050  
POLSR051  
POLSR052  
POLSR053  
POLSR054  
POLSR055  
POLSR056  
POLSR057  
POLSR058  
POLSR059  
POLSR060  
POLSR061

```

244 CONTINUE          POLSR062
    GO TO 145          POLSR063
245 CALL STPOL1       POLSR064
145 CCNTINUE          POLSR065
146 DC 150 I=1,18     POLSR066
150 SORC(I) = 0.      POLSR067
    SORC(10) = -1.     POLSR068
C
C   ENVRCN AREAS      POLSR069
C
C   IF (NENAR .EQ. 0) GO TO 156  POLSR070
    ITYPE = 1           POLSR071
    DO 155 I=1,NENAR   POLSR072
    DO 153 J=1,5       POLSR073
153 SORC(J) = ENAR(J,I)  POLSR074
    DO 154 J=6,11      POLSR075
154 SORC(J+5) = ENAR(J,I)  POLSR076
    DO 253 J=11,16     POLSR077
    IF (SORC(J).NE.0.0) GO TO 254  POLSP078
253 CCNTINUE          POLSR079
    GO TO 155          POLSR080
254 CALL SIPOL1       POLSR081
155 CCNTINUE          POLSP082
C
C   AIREASE AREAS      POLSR083
C
C   IF (NABAR .EQ. 0) GO TO 166  POLSR084
    ITYPE = 2           POLSP085
    DO 165 I=1,NABAR   POLSR086
    DC 163 J=1,5       POLSP087
163 SORC(J) = ABAR(J,I)  POLSP088
    DO 164 J=6,11      POLSR089
164 SORC(J+5) = ABAR(J,I)  POLSR090
    DO 263 J=11,16     POLSR091
    IF (SORC(J).NE.0.0) GO TO 264  POLSR092
263 CCNTINUE          POLSP093
    GO TO 165          POLSR094
264 CALL SIPOL1       POLSR095
165 CCNTINUE          POLSP096
C
C   AIRCRAFT AREAS      POLSR097
C
C   IF (NACAR .EQ. 0) GO TO 176  POLSP098
    ITYPE = 3           POLSR099
    DC 175 I=1,NACAR   POLSP100
    DO 173 J=1,5       POLSR101
173 SORC(J) = ACAR(J,I)  POLSP102
    DO 174 J=6,11      POLSR103
174 SORC(J+5) = ACAR(J,I)  POLSR104
    DO 273 J=11,16     POLSP105
    IF (SORC(J).NE.0.0) GO TO 274  POLSR106
273 CCNTINUE          POLSR107
    GO TO 175          POLSR108
274 CALL STPOL1       POLSR109
175 CCNTINUE          POLSP110
C
C   CRITICAL DISTANCE, XZ, MUST BE CONVERTED TO METERS FOR LINE MODEL  POLSR110
C
C   176 XZ = XZ * 1000.          POLSP111
C
C   ENVIRON LINES          POLSR112
C

```

IF (NENLN .EQ. 0) GO TO 186	POLSR124
ITYPE = 1	POLSR125
DC 185 I=1, NENLN	POLSP126
DC 184 J=1, 8	POLSR127
184 SOFC (J) = ENLN (J,I)	POLSR128
DO 384 J=13, 18	POLSR129
384 SCRC (J) = ENLN (J-4,I)	POLSR130
DO 284 J=13, 18	POLSR131
IF (SCRC (J) .NE. 0.0) GO TO 285	POLSR132
284 CONTINUE	POLSR133
GC TC 185	POLSR134
285 SCRC (9) = 100.	POLSR135
SORC (10) = 100.	POLSR136
SCRC (12) = 1.	POLSR137
CALL STPOL2	POLSR138
185 CONTINUE	POLSR139
C	POLSR140
C AIRBASE LINES	POLSR141
C	POLSR142
186 IF (NABIN .EQ. 0) GO TO 196	POLSR143
ITYPE = 2	POLSR144
DC 195 I=1, NABLN	POLSR145
DC 194 J=1, 8	POLSR146
194 SCRC (J) = ABLN (J,I)	POLSR147
DC 394 J=13, 18	POLSR148
394 SCRC (J) = ABLN (J-4,I)	POLSR149
DO 294 J=13, 18	POLSP150
IF (SOFC (J) .NE. 0.0) GO TO 295	POLSR151
294 CCNTINUE	POLSR152
GC TC 195	POLSP153
295 SCRC (9) = 100.	POLSR154
SCPC (10) = 100.	POLSR155
SCRC (12) = 1.	POLSR156
CALL STPOL2	POLSR157
195 CONTINUE	POLSR158
C	POLSR159
C AIRCRAFT LINES	POLSR160
C	POLSR161
196 IF (NACLN .EQ. 0) GO TO 206	POLSR162
ITYPE = 3	POLSP163
DO 205 I=1, NACLN	POLSR164
DO 204 J=1, 18	POLSR165
204 SCRC (J) = ACLN (J,I)	POLSR166
DC 304 J=13, 18	POLSR167
IF (SCRC (J) .NE. 0.0) GO TO 305	POLSP168
304 CCNTINUE	POLSR169
GO TC 205	POLSP170
305 CALI STPOL2	POI SR171
205 CONTINUE	POLSR172
206 CCNTINUE	POLSR173
RETURN	POLSR174
END	POLSR175

## SUBROUTINE PSEUDO

### Purpose:

To call the SIGCY and SIGCZ functions to find the virtual distance in meters from the source to the pseudo upwind point source.

### Input:

1. Initial dispersions in y and z directions.
2. Wind speed and stability class.

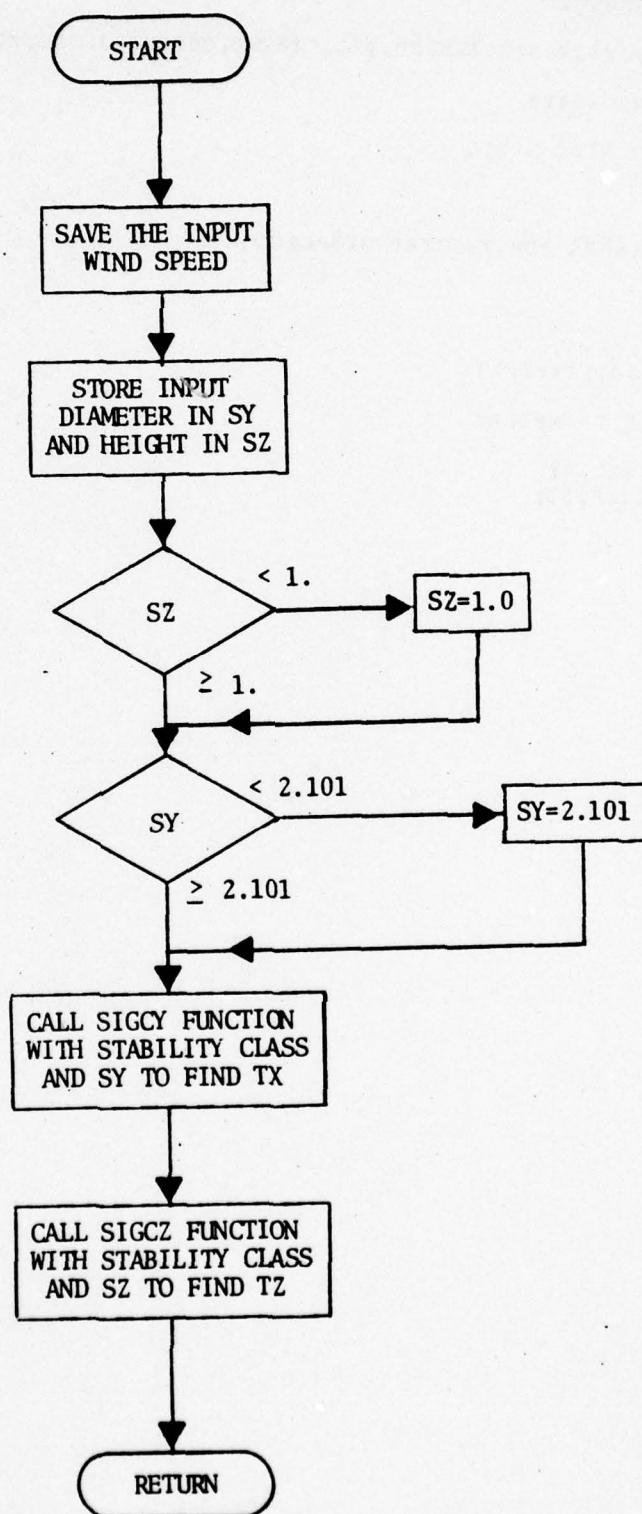
### Output:

The virtual distances in the y and z directions.

### Functions Called:

SIGCY  
SIGCZ

SUBROUTINE PSEUDO



SUBROUTINE PSEUDO (DS,WIN,HS,TY,TZ)	PSUDO000
C	PSUDO001
C THIS SUBROUTINE CALLS THE SIGCY AND SIGCZ FUNCTIONS	PSUDO002
C TO FIND THE VIRTUAL DISTANCE FROM THE SOURCE TO THE PSEUDO	PSUDO003
C UPWIND POINT SOURCE	PSUDO004
C	PSUDO005
COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,	PSUDO006
• TEMP,TEMK	PSUDO007
COMMON /WDUN/ WSAVE	PSUDO008
C	PSUDO009
C SAVE THE INPUT WIND SPEED	PSUDO010
C	PSUDO011
WSAVE=WIN	PSUDO012
C	PSUDO013
C SET MINIMUM VALUES FOR INITIAL DISPERSIONS	PSUDO014
C	PSUDO015
SY=DS	PSUDO016
SZ=HS	PSUDO017
IF (SZ.LT.1.) SZ=1.0	PSUDO018
IF (SY.LT.2.101) SY=2.101	PSUDO019
C	PSUDO020
C FIND DISTANCES IN METERS	PSUDO021
C	PSUDO022
TY=SIGCY (JSTAR,SY)	PSUDO023
TZ=SIGCZ (JSTAB,SZ)	PSUDO024
RETURN	PSUDO025
END	PSUDO026

## SUBROUTINE QMOD

### Purpose:

To compute the linear distribution, in inverse length, of the pollution along a runway due to aircraft emission during landing or takeoff.

### Input:

YSI Distance along runway measured from tip of exhaust plume near starting end of runway  
TAIL Length or penetration of exhaust plume of aircraft at rest  
DL Length of smoke slug on runway  
A Acceleration (or deceleration) of aircraft  
V12 Initial velocity squared  
VS Average velocity of exhaust particles relative to air mass in exhaust plume  
WS2 Wind speed squared  
WSC 2·wind speed·(- cosine of angle between runway and wind vector)  
RR A/G, where A is acceleration and G is the normalization constant for line density

### Output:

QL The linear distribution of pollution

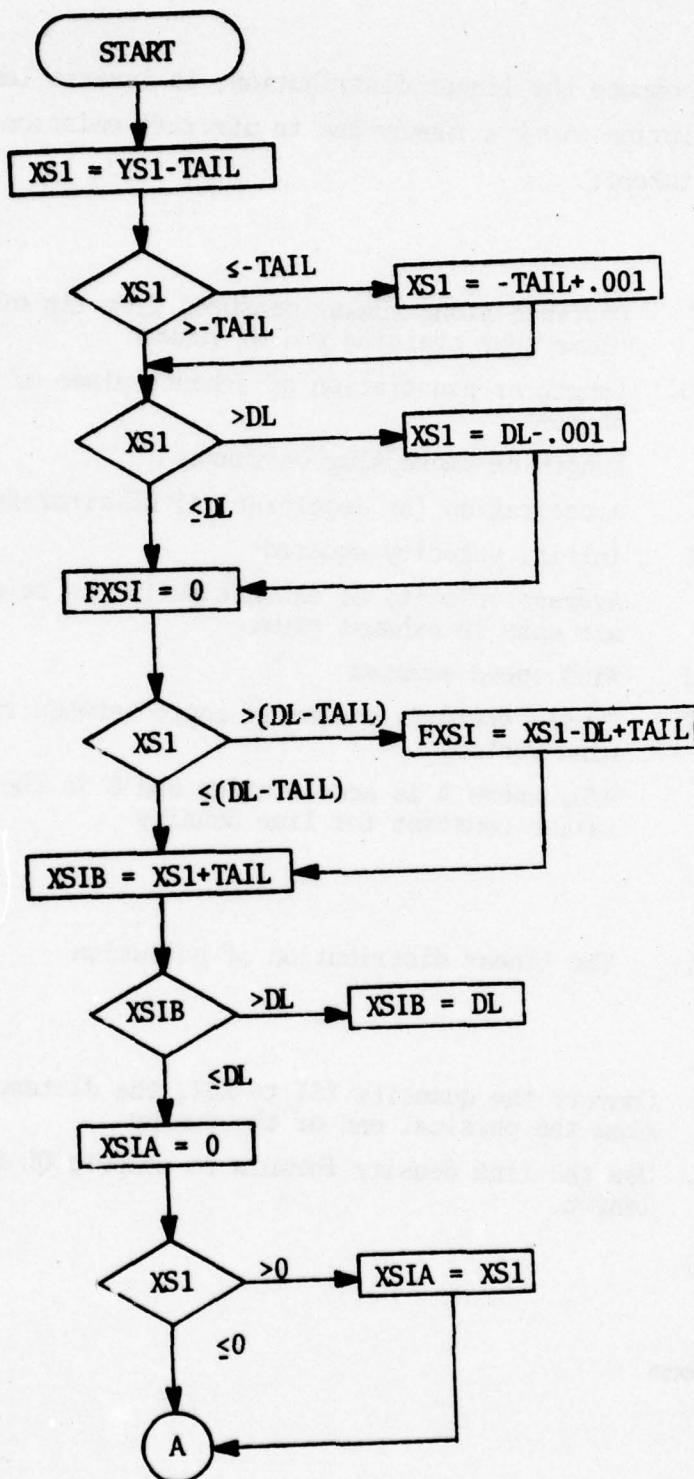
### Procedure:

1. Convert the quantity YSI to XSI, the distance measured from the physical end of the runway.
2. Use the line density formula to compute QL in inverse length.

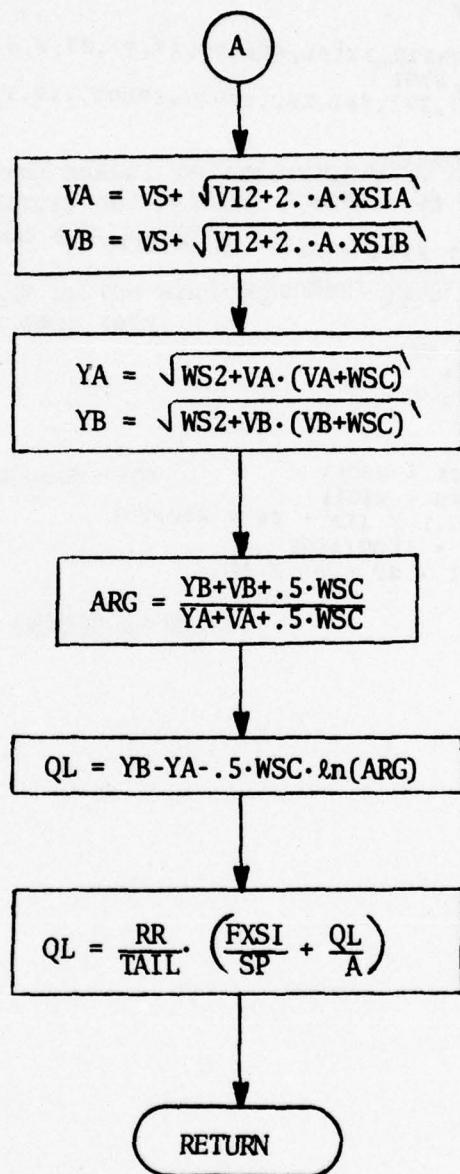
### Subroutines Called:

None

SUBROUTINE QMOD



SUBROUTINE QMOD (Contd.)



```

SUBROUTINE QMOD (YS1,QL) QMOD0000
C QMOD0001
C THIS ROUTINE COMPUTES THE LINEAR DISTRIBUTION, IN INVERSE LENGTH, QMOD0002
C OF THE POLLUTION ALONG A RUNWAY DUE TO AIRCRAFT EMMISION QMOD0003
C DURING LANDING OR TAKEOFF QMOD0004
C QMOD0005
COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAZERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,QMOD0006
. V1,V2,DL,TIME,FMIS(6),NPOL QMOD0007
COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,QMOD0008
. WS2,WSC,RR,SP QMOD0009
XS1 = YS1 - TAIL QMOD0010
IF (XS1 .LE. -TAIL) XS1 = -TAIL + .001 QMOD0011
IF (XS1 .GT. DL) XS1 = DL - .001 QMOD0012
FXSI = 0. QMOD0013
IF (XS1 .GT. (DL-TAIL)) FXSI = XS1 - DL + TAIL QMOD0014
30 XSIB = XS1 + TAIL QMOD0015
IF (XSIB .GT. DL) XSIB = DL QMOD0016
XSIA = 0. QMOD0017
IF (XS1 .GT. 0) XSIA = XS1 QMOD0018
ROOTB = V12 + 2.*A*XSIB QMOD0019
ROOTA = V12 + 2.*A*XSIA QMOD0020
VA = SQRT(ROOTA) + VS QMOD0021
VB = SQRT(ROOTB) + VS QMOD0022
YA = SQRT(WS2 + VA *(VA + WSC)) QMOD0023
YB = SQRT(WS2 + VB *(VB + WSC)) QMOD0024
ARG = (YB + VB + WSC/2.) / (YA + VA + WSC/2.) QMOD0025
QL = YB - YA - WSC/2. * ALOG(ARG) QMOD0026
QL = RR / TAIL * (FXSI / SP + QL / A) QMOD0027
RETURN QMOD0028
END QMOD0029

```

## SUBROUTINE READ

### Purpose:

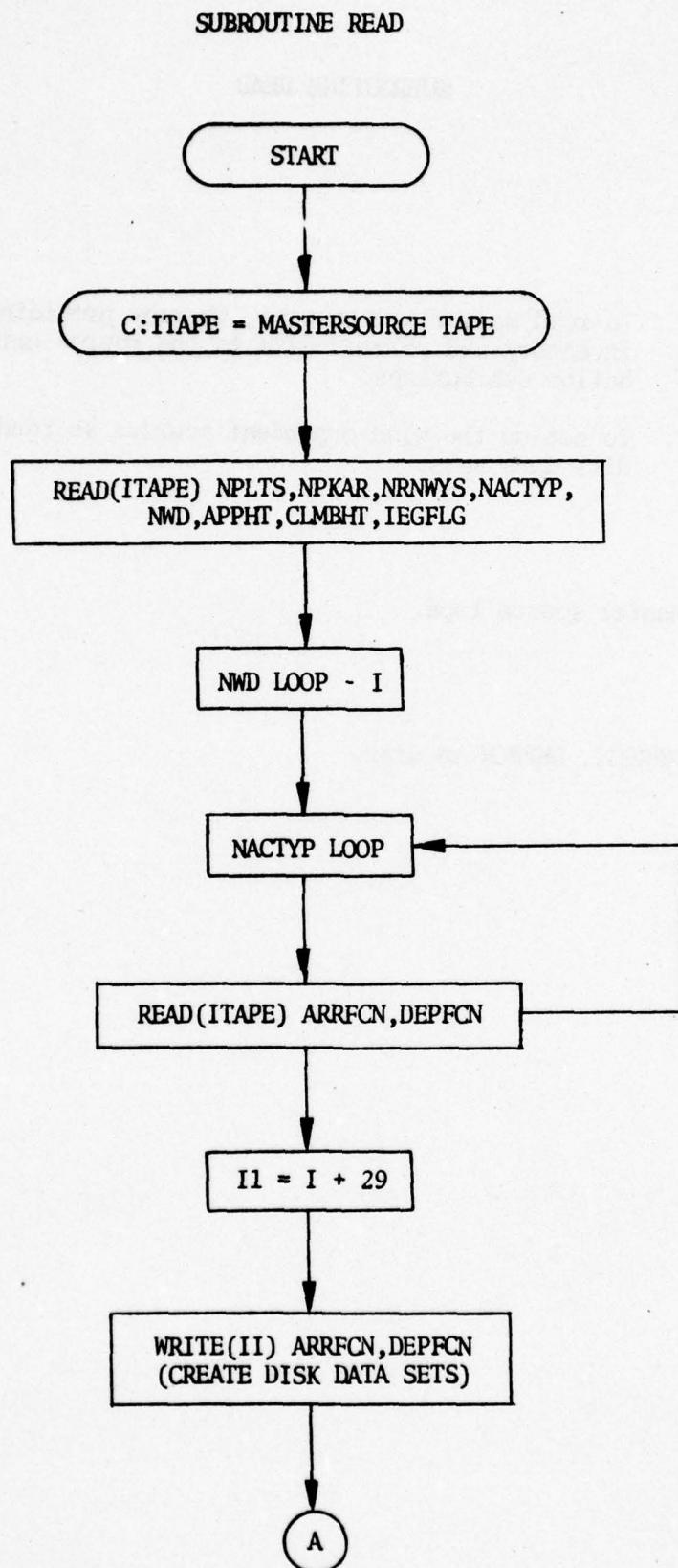
1. To read master source tape, thereby providing the emission inventory and related data to the source emission distribution subroutines.
2. To set up the wind-dependent sources as random access disk data sets.

### Input:

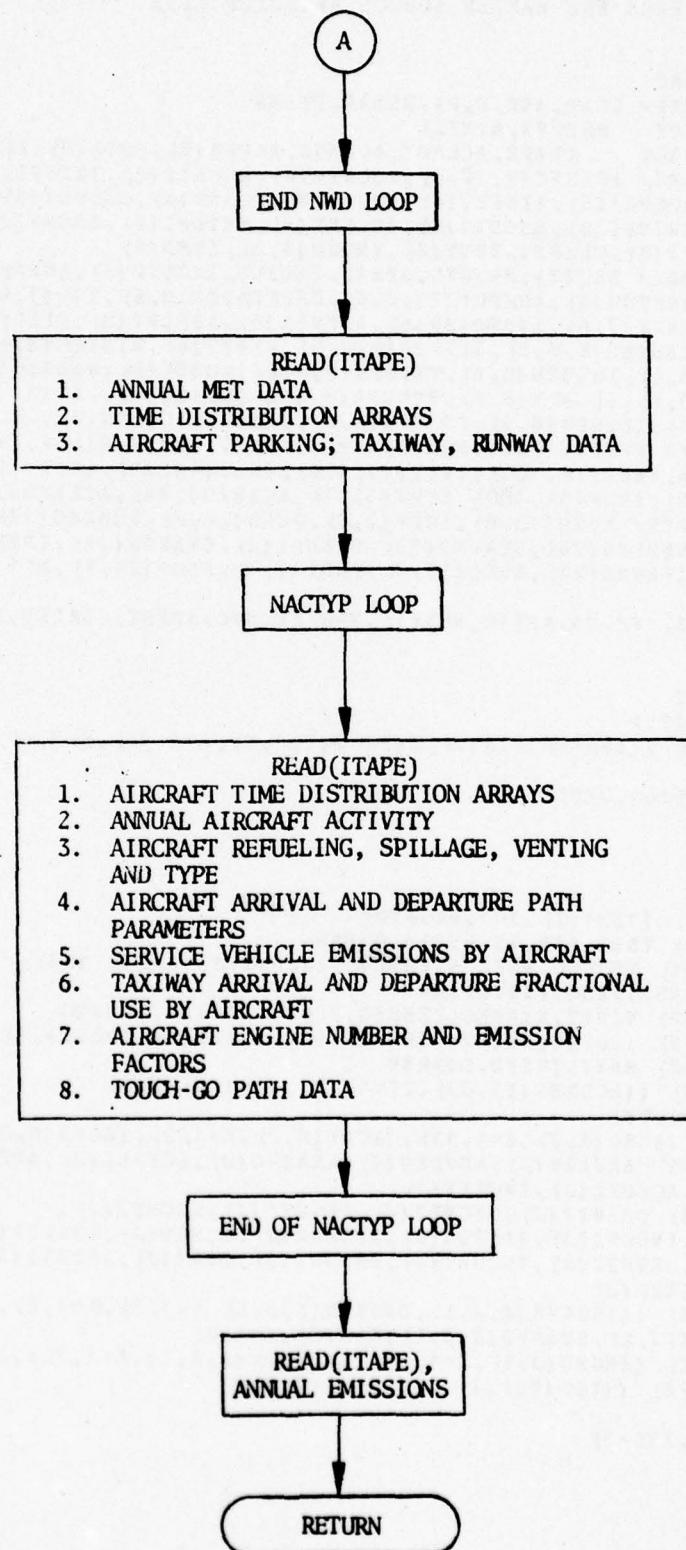
Master source tape.

### Output:

ARRFCN, DEPFCN to disk.



SUBROUTINE READ (Continued)



```

SUBROUTINE READ                                     READ0000
C THIS ROUTINE READS THE MASTER SOURCE EMISSION TAPE   READ0001
C
REAL LNDSPD                                     READ0002
INTEGER ENGNO                                    READ0003
COMMON /ANNMET/ TBAR, ADD, E, PA, WSBAR, DTBAR   READ0004
COMMON /RECPT/ MRECPT, MAXFIL                   READ0005
COMMON /DEFAUT/ ITAPE, ACLNDY, ACLNDZ, ALPHA(7), BETA(7), F1DENS(7) READ0006
COMMON /ACEDB1/ ACEMFC(8,10,6), ASCNT1(8), ASCNT2(8), TXISPD(8),   READ0007
. LNDSED(8), APSPD1(8), APSPD2(8), COHT1(8), TOSPD(8), COSPD1(8),   READ0008
. COSED2(8), SRTUPT(8), DSCNT1(8), EGCHK(8), SHTDNT(8), DSCNT2(8),   READ0009
. APPHT, APPHT2(8), CLMBHT, TOWT(8), ENGNO(8,2), IDR(8)           READ0010
COMMON /ACEDE2/ NACTYF, NRNWYS, NPKAR, IEGFLG, IACTYP(8), ANNARR(8),   READ0011
. ANNDEP(8), ANNTGO(8), ARRFCN(24,8,6), DEPFCN(24,8,6), TGO(3,4,8),   READ0012
. DISRNW(6), RNWY(7,6), IUSWD(20,6), ACFUEL(8), ARFLVT(8), DPFLVT(8),   READ0013
. ACSFIL(8), ARSLEM(6,8,5), DPSVEM(6,8,5), NIBTT(6), NIBSEG(8,6),   READ0014
. IIBSEG(16,8,6), IDIBTW(8,6), TTARFR(8,8,6), NOBTT(6), NOBSEG(8,6),   READ0015
. IOBSEG(16,8,6), IDOBTW(8,6), TTDPFR(8,8,6), NPASQ(6), IDPRKA(6),   READ0016
. PAREA(6,3,3), IDIBPA(8,6), IDOBPA(8,6), NLSEGS, ACLNSG(12,25), JES1(8) READ0017
COMMON /SRCE/ NPLTS, NENPT, NENAR, NENLN, NABPT, NABAR, NABL, NACPT,   READ0018
. NACAR, NACLN, ENFT(16,100), ENAR(11,100), ENLN(14,20), ABPT(16,150),   READ0019
. ABAR(11,100), ABLN(14,100), ACPT(16,1), ACAR(11,24), ACLN(18,25)   READ0020
COMMON /DSTRET/ ACMO(13,8), ACDY(2,8), ACHR(24,8), VHMLMO(13),   READ0021
. VHMLDY(2), VHMLHR(24), CVABMO(13), CVABDY(2), CVABHR(24), CVENMO(13),   READ0022
. CVENDY(2), CVENHR(24), FLMO(13,7), FLDY(2,7), FLHR(24,7), NC1   READ0023
C
REAL (ITAPE) NPLTS, NPKAR, NRNWYS, NACTYP, NWD, APPHT, CLMBHT, IEGFLG   READ0024
. , NLSEGS                                         READ0025
REWIND 30                                         READ0026
DC 2 I=1, NWD                                     READ0027
DC 5 J=1, NACTYP                                   READ0028
READ (ITAPE) ((ARRFCN(L,J,K), DEPFCN(L,J,K), L=1,24), K=1,6)   READ0029
5 CCNTINUE                                         READ0030
WRITE(30) ARRFCN, DEPFCN                         READ0031
2 CCNTINUE                                         READ0032
REWIND 30                                         READ0033
MRECPT=1                                         READ0034
MAXFIL=NWD                                       READ0035
READ (ITAPE) (JES1(I), I=1, NACTYP)               READ0036
READ (ITAPE) TBAR, ADD, EA, WSBAR, DTBAR         READ0037
READ (ITAPE) VHMLMO, VHML, NIBTT, NIBSEG, IOBSEG   READ0038
. CVENDY, CVENHR, FLMO, FLDY, FLHR             READ0039
READ (ITAPE) NIBTT, NIBSEG, IIBSEG, NOBTT, NOBSEG, IOBSEG   READ0040
READ (ITAPE) IDOBTW, IDIBTW, IDPRKA, PAREA, IDIBPA, IDOBPA, NPASQ   READ0041
READ (ITAPE) RNWY, IUSWD, DISRNW                 READ0042
READ (ITAPE) ((ACLNSG(II,JJ), II=1,12), JJ=1, NLSEGS)   READ0043
DC 40 J=1, NACTYP                                 READ0044
READ (ITAPE) (ACMO(K,J), K=1,13), (ACDY(K,J), K=1,2), (ACHR(K,J), K=1,24) READ0045
READ (ITAPE) ANNARR(J), ANNDEP(J), ANNTGO(J), ACFUEL(J), ARFLVT(J),   READ0046
. DPFLVT(J), ACSPIL(J), IACTYP(J)               READ0047
READ (ITAPE) DSCNT1(J), DSCNT2(J), ASCNT1(J), ASCNT2(J),   READ0048
. TXISPD(J), LNDSPD(J), APSPD1(J), APSPD2(J), TOSPD(J), COSED1(J),   READ0049
. COSED2(J), SRTUPT(J), EGCHK(J), SHTDNT(J), TOWT(J), APPHT2(J),   READ0050
. COHT1(J), IDRF(J)                           READ0051
READ (ITAPE) ((ARSLEM(K,J,L), DPSVEM(K,J,L), L=1,5), K=1,6),   READ0052
. ((TTARFR(K,J,L), TTDPFR(K,J,L), K=1,8), L=1,6)           READ0053
READ (ITAPE) (ENGNO(J,I), I=1,2), ((ACEMFC(J,K,L), K=1,10), L=1,6)   READ0054
READ (ITAPE) ((TGO(K,L,J), K=1,3), L=1,4)           READ0055
40 CCNTINUE                                         READ0056
4 READ (ITAPE, END=3)                           READ0057
GO TO 4                                         READ0058
                                         READ0059
                                         READ0060
                                         READ0061

```

3 CCNTINUE  
RETURN  
END

READ0062  
READ0063  
READ0064

## FUNCTION RISE

### Purpose:

To calculate the plume rise using either the Carson-Moses or Holland plume rise formula.

### Input:

Stack parameters, current wind speed and stability, temperature, and the plume rise flag.

### Output:

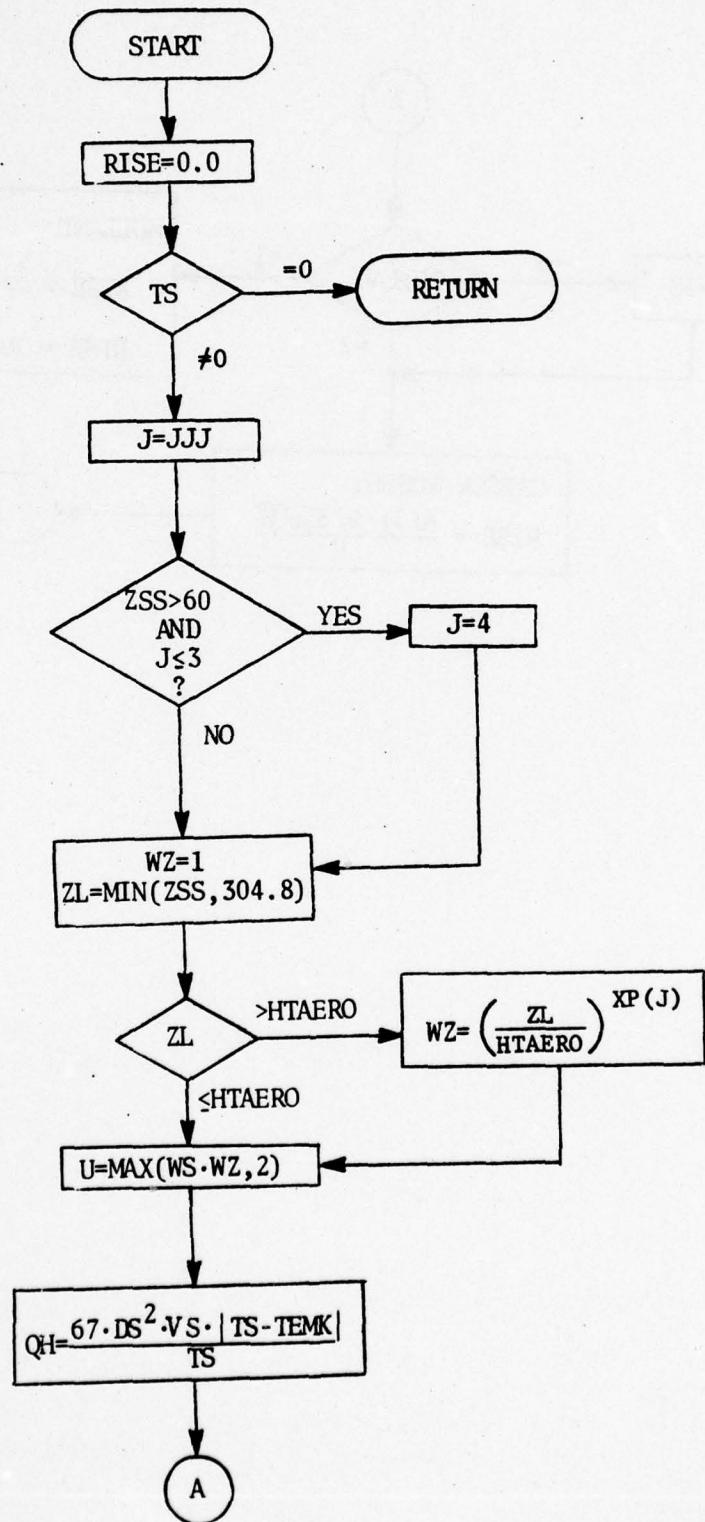
The height of the plume rise.

### Subroutines

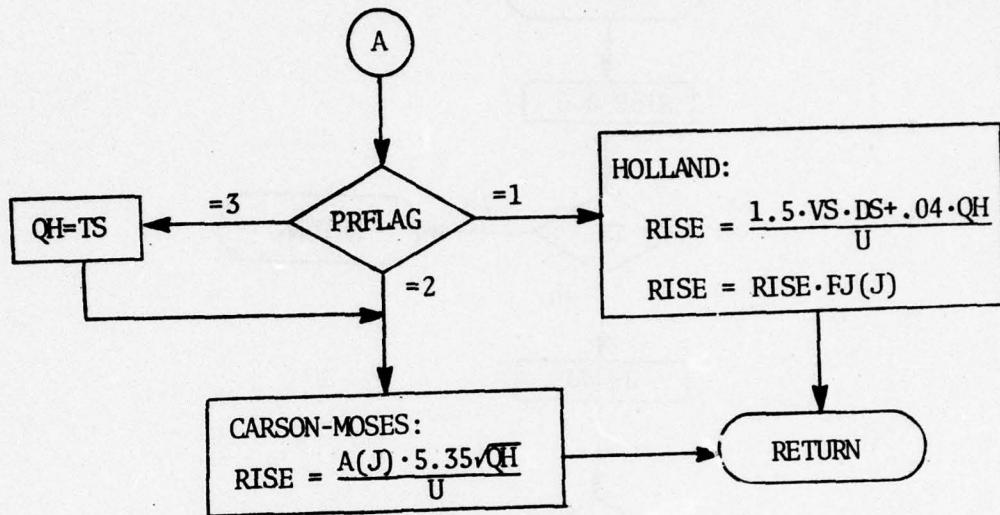
#### Called:

None

FUNCTION RISE



FUNCTION RISE (Cont'd.)



```

FUNCTION RISE(ZSS, JJJ) RISE0000
C RISE0001
C THIS FUNCTION CALCULATES THE PLUME RISE RISE0002
C ZSS IS THE PHYSICAL STACK HEIGHT MODIFIED FOR DOWNWASH RISE0003
C EFFECTS, IF ANY RISE0004
C JJJ IS THE AMBIENT STABILITY RISE0005
C RISE0006
C RISE0007
COMMON /MET/ WS, WSMPH, IWS, WD, IWD, SINEWD, COSEWD, JSTAB, HLID, TEMP, RISE0008
. TEMK
COMMON /INFO/ IRECEP, IWNDIR, ITYPE, HTAERO, XS, YS, ZS, DELY, DELZ, RISE0009
. TS, VS, DS, HB, PRFLAG, EMIS(8), NPOL RISE0010
DIMENSION A(6), FJ(6) RISE0011
COMMON /WNDPRO/ XP(6) RISE0012
DATA A /2.65, 2.65, 2.65, 1.08, 2*0.68/, RISE0013
. FJ / 1.2, 1.2, 1.2, 1.0, 0.8, 0.8/ RISE0014
RISE=0.0 RISE0015
C RISE0016
C CHECK THE STACK EXIT GAS TEMPERATURE RISE0017
C RISE0018
IF (TS.EQ.0.0) RETURN RISE0019
C RISE0020
FOR TALL STACKS USE STABILITY 4 IN THE WIND PROFILE LAW RISE0021
C RISE0022
J=JJJ RISE0023
IF (ZSS.GT.60. AND. J.LE. 3) J=4 RISE0024
C RISE0025
C COMPUTE THE WIND SPEED AT THE ELEVATION OF THE STACK RISE0026
C FOR STABILITY J RISE0027
C RISE0028
WZ=1.0 RISE0029
ZL=AMIN1(ZSS, 304.8) RISE0030
IF (ZL.GT. HTAERO) WZ= (ZL/HTAERO) **XP(J) RISE0031
U=AMAX1(WS*WZ, 2.0) RISE0032
C RISE0033
C COMPUTE THE THERMAL EMISSION RATE RISE0034
C RISE0035
QP=67.0*DS*DS*VS*ABS(TS-TEMK)/TS RISE0036
IF (PRFLAG.EQ.1.0) GO TO 1 RISE0037
IF (PRFLAG.EQ.3.0) QH=TS RISE0038
C RISE0039
C CARSON-MOSES PLUME RISE FORMULA RISE0040
C RISE0041
RISE=A(J) *5.35*SQRT(QH)/U RISE0042
RETURN RISE0043
C RISE0044
C HOLLAND PLUME RISE FORMULA RISE0045
C RISE0046
1 CONTINUE RISE0047
RISE=1.5*VS*DS/U+0.04*QH/U RISE0048
RISE=RISE*FJ(J) RISE0049
RETURN RISE0050
END RISE0051

```

FUNCTION RRDIST

Purpose:

To calculate the length of runway necessary for takeoff using aircraft dependent equations.

Input:

Aircraft identification, pressure altitude, ambient temperature and wind speed, and aircraft takeoff weight.

Output:

Takeoff length in feet of runway roll to liftoff.

Procedure:

For a given aircraft, use the proper set of takeoff equations provided by the USAF.

Subroutines

Called:

None

```

FUNCTION RRDIST (IR,PA,T,GW,WS) RRDST000
C RRDST001
C FUNCTION CALCULATES RUNWAY ROLL DISTANCE IN FEET RRDST002
C IR IS AIRCRAFT IDENTIFICATION NUMBER RRDST003
C PA IS PRESSURE ALTITUDE IN HUNDREDS OF FEET RRDST004
C T IS TEMPERATURE IN DEGREES FAHRENHEIT RRDST005
C GW IS AC TAKE OFF WEIGHT IN THOUSAND POUNDS RRDST006
C WS IS THE WIND SPEED IN KNOTS RRDST007
C RRDST008
C FGR=0.0 RRDST009
IF (IR.EQ.100) GO TO 100 RRDST010
GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22, RRDST011
123,24,25,26,27,28,29,30,31,32,33,34,35,36,37,100,100,100,100,100, RRDST012
2 100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100, RRDST013
1 CCNTINUE RRDST014
GC TO 100 RRDST015
3 CCNTINUE RRDST016
2 TOF=-(2.78-8.5714E-4*PA)+(1.82E-2+7.2857E-5*PA)*GW RRDST017
GR=(1.184E+1-4.2167E-1*T+1.0E-2*T**2-4.583E-5*T**3)+ RRDST018
. (4.194+1.7197E-2*T-9.26018E-4*T**2)*TOF+ RRDST019
. (1.0457+8.40E-3*T+2.117E-4*T**2+2.98E-7*T**3)*TOF**2 RRDST020
FGR=(GR-(1.15E-1+9.0E-3*GR)*WS)*100. RRDST021
GO TO 100 RRDST022
4 CCNTINUE RRDST023
5 TOF=(1.589+6.883E-3*PA+1.2767E-4*PA**2)+ RRDST024
. (6.819E-3+1.1007E-4*PA-3.924E-7*PA**2)*T+ RRDST025
. (5.979E-5+3.38096E-7*PA+8.532E-9*PA**2)*T**2 RRDST026
GR=(-13.25+8.75E-1*GW-1.25E-2*GW**2)+ RRDST027
. (1.3925E+1-9.275E-1*GW+2.125E-2*GW**2)*TOF RRDST028
FGR=(GR-(1.316E-1+8.748E-3*GR)*WS)*100. RRDST029
GC TO 100 RRDST030
6 TOF=(9.3937E-1+2.0947E-2*PA+2.005E-4*PA**2)+ RRDST031
. (3.746467E-2+4.05625E-4*PA)*T+ RRDST032
. (1.9928E-4-5.75006E-6*PA+1.40234E-7*PA**2)*T**2 RRDST033
GR=(1.4307E+1-7.57144E-1*GW+2.6785E-2*GW**2)+ RRDST034
. (1.67257E+1-1.17762*GW+2.7381E-2*GW**2)*TOF RRDST035
FGR=(GR-(2.412799E-2+7.82971E-3*GR)*WS)*100. RRDST036
GO TC 100 RRDST037
7 TOF=(-1.06E-3+1.674E-2*PA+8.1888E-5*PA**2)+ RRDST038
. (1.36E-2+9.592E-6*PA+1.755E-6*PA**2)*T+ RRDST039
. (5.1099E-5+1.2899E-6*PA-6.123E-9*PA**2)*T**2 RRDST040
GR=(-1.423E+1+6.349998E-1*GW+1.6667E-3*GW**2)+ RRDST041
. (6.1857-3.2179E-1*GW+8.214E-3*GW**2)*TOF RRDST042
FGR=(GR-(6.293E-2+7.328E-3*GR)*WS)*100. RRDST043
GO TC 100 RRDST044
8 TOF=(9.503E-2+3.313E-2*PA+1.3666E-4*PA**2)+ RRDST045
. (2.2546E-2+1.7848E-4*PA-4.04E-6*PA**2)*T+ RRDST046
. (1.3438E-4-1.2166E-6*PA+4.1854E-8*PA**2)*T**2 RRDST047
GR=(2.95E+1-2.394*GW+6.497E-2*GW**2)+ RRDST048
. (3.1035+7.52E-2*GW-3.186E-3*GW**2)*TOF+ RRDST049
. (1.2715-1.5535E-1*GW+4.3889E-3*GW**2)*TOF**2 RRDST050
FGR=(GR-(-9.0E-2+1.807E-2*GR-7.143E-5*GR**2)*WS)*100. RRDST051
GO TO 100 RRDST052
9 TOF=(3.36455E-3+5.63556E-2*PA)+ RRDST053
. (4.417E-2-2.031E-3*PA+5.63E-5*PA**2-3.9954E-7*PA**3)*T+ RRDST054
. (-9.2E-5+2.08E-5*PA-5.39E-7*PA**2+3.8E-9*PA**3)*T**2 RRDST055
GR=(1.65838-3.069E-1*GW+8.1363E-2*GW**2)+ RRDST056
. (-3.6111+3.63559E-1*GW)*TOF+ RRDST057
. (7.3975E-1-8.78749E-2*GW+3.2487E-3*GW**2)*TOF**2 RRDST058
FGR=(GR-(5.0E-2+7.4E-3*GR)*WS)*100. RRDST059
GC TC 100 RRDST060
10 TOF=(12.5546-5.7192E-2*PA+1.3075E-4*PA**2)- RRDST061

```

. (2.9032E-2-1.0254E-4\*PA-1.45125E-7\*PA\*\*2) \*T RRDST062  
 GR=((-5.14955E+1+2.57957\*GW-1.4425E-2\*GW\*\*2) - RRDST063  
 . (-1.1535E+1+5.915E-1\*GW-4.6828E-3\*GW\*\*2) \*TOF+ RRDST064  
 . (-6.2285E-1+3.2375E-2\*GW-2.9056E-4\*GW\*\*2) \*TOF\*\*2) \*1000. RRDST065  
 FGR=(3.305E+1+9.729E-1\*GR+2.31E-6\*GR\*\*2) - RRDST066  
 . (8.244+8.3598E-3\*GR-1.44E-8\*GR\*\*2) \*WS RRDST067  
 GO TO 100 RRDST068  
 11 TCF=(7.436E-1+4.29E-2\*PA)+(2.1276E-2-3.1116E-5\*PA)\*T RRDST069  
 GR=(1.638E+1-7.78E-1\*GW+2.84E-2\*GW\*\*2)+ RRDST070  
 . (3.809-1.947E-1\*GW+4.264E-3\*GW\*\*2) \*TOF+ RRDST071  
 . (-1.976E-1+1.5757E-2\*GW+4.6189E-4\*GW\*\*2) \*TOF\*\*2 RRDST072  
 FGR=(GR-(8.5E-2+8.25E-3\*GR)\*WS)\*100. RRDST073  
 GC TO 100 RRDST074  
 12 TCF=(1.1405-4.659E-3\*PA+1.28E-5\*PA\*\*2)- RRDST075  
 . (2.0146E-3-2.46E-5\*PA+3.5514E-7\*PA\*\*2) \*T RRDST076  
 GR=(-3.0029E+1-9.6225E-2\*GW+1.25428E-1\*GW\*\*2)- RRDST077  
 . (-7.3845E+1+1.20433\*GW+1.7857E-1\*GW\*\*2) \*TOF+ RRDST078  
 . (-3.57857E+1+7.857E-1\*GW+7.14286E-2\*GW\*\*2) \*TOF\*\*2 RRDST079  
 FGR=((3.17413E-1+9.762E-1\*GR+2.657E-4\*GR\*\*2)- RRDST080  
 . (1.1114E-1+7.91177E-3\*GR+4.40169E-5\*GR\*\*2) \*WS)\*100. RRDST081  
 GC TO 100 RRDST082  
 13 TOF=(9.166-5.485E-2\*PA)-(3.412E-2-1.8E-4\*PA)\*T RRDST083  
 GR=(3.02E+2-3.519E+1\*GW+1.841\*GW\*\*2)- RRDST084  
 . (1.306E+2-1.277E+1\*GW+5.4E-1\*GW\*\*2) \*TOF+ RRDST085  
 . (2.0687E+1-1.715\*GW+6.07E-2\*GW\*\*2) \*TOF\*\*2- RRDST086  
 . (1.1578-8.4228E-2\*GW+2.46E-3\*GW\*\*2) \*TOF\*\*3 RRDST087  
 FGR=(GR-(9.55E-2+7.15E-3\*GR)\*WS)\*100. RRDST088  
 GO TO 100 RRDST089  
 14 TOF=(2.336+1.582E-2\*PA+1.172E-4\*PA\*\*2)+ RRDST090  
 . (5.604E-3+9.97746E-5\*PA-5.8117147E-7\*PA\*\*2) \*T+ RRDST091  
 . (9.19269E-5-1.34357E-8\*PA+1.61411E-8\*PA\*\*2) \*T\*\*2 RRDST092  
 GR=(7.7366-2.52997E-1\*GW+2.385E-3\*GW\*\*2)+ RRDST093  
 . (-2.1071+4.2586E-2\*GW+12.748E-4\*GW\*\*2) \*TOF RRDST094  
 FGR=(GR-(1.0755E-1+1.4588E-2\*GR-7.94156E-5\*GR\*\*2)\*WS)\*100. RRDST095  
 GC TO 100 RRDST096  
 15 CONTINUE RRDST097  
 GC TO 100 RRDST098  
 16 TOF=(7.6859-1.15E-1\*PA+4.413E-4\*PA\*\*2)- RRDST099  
 . (2.925E-2-8.1128E-4\*PA+6.999E-6\*PA\*\*2) \*T- RRDST100  
 . (2.2289E-4+5.054E-6\*PA-7.57E-8\*PA\*\*2) \*T\*\*2 RRDST101  
 GR=(2.546E+1-2.3388\*GW+1.0717E-1\*GW\*\*2)- RRDST102  
 . (7.9095-6.7434E-1\*GW+2.1045E-2\*GW\*\*2) \*TOF+ RRDST103  
 . (6.099E-1-5.0858E-2\*GW+1.434E-3\*GW\*\*2) \*TOF\*\*2 RRDST104  
 FGR=(GR-(1.16E-1+7.27E-3\*GR-3.64E-6\*GR\*\*2)\*WS)\*100. RRDST105  
 GC TO 100 RRDST106  
 17 CCNTINUE RRDST107  
 GC TO 100 RRDST108  
 18 TOF=(2.118+1.058E-2\*PA+1.014E-4\*PA\*\*2)+ RRDST109  
 . (2.102E-3+1.84E-4\*PA-1.177E-6\*PA\*\*2) \*T+ RRDST110  
 . (1.001E-4-7.046E-7\*PA+1.355E-8\*PA\*\*2) \*T\*\*2 RRDST111  
 GR=(1.0E-5)+(-1.9687+4.209E-1\*GW+3.9445E-2\*GW\*\*2) \*TOF RRDST112  
 FGR=(GR-(8.363E-2+1.488E-2\*GR-9.78E-5\*GR\*\*2)\*WS)\*100. RRDST113  
 GC TO 100 RRDST114  
 19 TCF=(4.65478+6.94444E-3\*T)+(3.257E-1+2.7778E-4\*T)\*(PA/10.) RRDST115  
 GR=(.1457+3.5625E-2\*GW-6.763E-5\*GW\*\*2)+ RRDST116  
 . (5.1428-3.175E-2\*GW+7.0089E-5\*GW\*\*2) \*TOF RRDST117  
 FGR=(GR-(.1+0.0082\*GR)\*WS)\*100. RRDST118  
 GC TO 100 RRDST119  
 20 TOF=(1.2192956+2.2091577E-3\*PA+3.380102E-4\*PA\*\*2)+ RRDST120  
 . (1.4628966E-2+2.6313968E-4\*PA-1.3818053E-7\*PA\*\*2) \*T- RRDST121  
 . (2.4891E-4-6.875E-6\*PA+7.8125E-8\*PA\*\*2) \*T\*\*2+ RRDST122  
 . (2.20314E-6-6.49E-8\*PA+7.47E-10\*PA\*\*2) \*T\*\*3 RRDST123

$GF = ((2.3806396 - 5.9265772E-2*GW + 6.67969E-4*GW**2) +$  RRDST124  
 $.- (-1.19933136 + 5.041098E-2*GW - 2.12517E-4*GW**2) * TOF) * 10.$  RRDST125  
 $FGF = (1.0 + 9.7757143E+1*GR + 6.4285714E-2*GR**2) -$  RRDST126  
 $. (4.6785706 + 5.4275515E-1*GR + 4.438775E-3*GR**2) * WS$  RRDST127  
 $GC TO 100$  RRDST128  
21  $TOF = (-4.799107E-1 + 3.3165178E-2*PA + 2.7902E-4*PA**2) +$  RRDST129  
 $. (2.129E-2 + 2.2538E-4 * PA - 2.9186E-6 * PA ** 2) * T$  RRDST130  
 $GR = (1.16103 + 5.318E-2 * GW + 9.0525E-4 * GW ** 2) +$  PRDST131  
 $. (3.3695E1 - 6.94278E-1 * GW + 3.8559E-3 * GW ** 2) * TOF -$  RRDST132  
 $. (-9.041 + 2.307E-1 * GW - 1.264E-3 * GW ** 2) * TOF ** 2 +$  PRDST133  
 $. (-1.0708 + 2.477E-2 * GW - 1.108E-4 * GW ** 2) * TOF ** 3$  PRDST134  
 $FGF = (GR - (2.4131E-1 + 2.115E-4*GR + 1.935E-4*GR**2) * WS) * 100.$  RRDST135  
 $GO TO 100$  RRDST136  
22 CCNTINUE RRDST137  
23  $TOF = (3.9116E-2 + 5.3976E-2*PA) + (1.6557E-2 - 7.6643E-6*PA) * T$  PRDST138  
 $GR = (5.625 - 9.5E-2*GW + 1.3125E-3*GW**2) +$  PRDST139  
 $. (8.6496E-1 - 1.2768E-2*GW + 1.077E-4*GW**2) * TOF +$  PRDST140  
 $. (4.0067E-1 - 5.382E-3*GW + 3.627E-5*GW**2) * TOF**2$  RRDST141  
 $FGF = (GR - (1.508E-1 + 8.625E-3*GR) * WS) * 100.$  RRDST142  
 $GC TO 100$  RFDST143  
24  $TOF = (5.4067E+1 - 1.3375E-1*PA - 2.2755E-4*PA**2 + 3.6508E-6*PA**3) -$  RRDST144  
 $. (7.395E-2 - 1.71E-4*PA - 5.91E-6*PA**2 + 4.22E-8*PA**3) * T$  RRDST145  
 $GR = (8.6549E+3 - 7.75196E+1*GW + 2.07846E-1*GW**2) -$  RFDST146  
 $. (5.6302E+2 - 4.9948*GW + 1.30519E-2*GW**2) * TOF +$  PRDST147  
 $. (1.22509E+1 - 1.07805E-1*GW + 2.759985E-4*GW**2) * TOF**2 -$  RRDST148  
 $. (8.8948E-2 - 7.77463E-4*GW + 1.956483E-6*GW**2) * TOF**3$  RRDST149  
 $FGF = (GR - (1.4123219E-1 + 8.5293578E-3*GR + 5.709895E-6*GR**2) * WS) * 100.$  RRDST150  
 $GC TO 100$  RRDST151  
25  $TOF = (7.90371 + 6.68965E-2*PA + 2.12622E-4*PA**2) +$  RRDST152  
 $. (3.00808E-2 + 2.67118E-5*PA + 9.85E-6*PA**2) * T +$  RRDST153  
 $. (1.23149E-4 + 1.3589E-6*PA - 3.1641E-8*PA**2) * T**2$  RRDST154  
 $GR = (2.1742857 + 2.04286E-1*GW - 1.071429E-2*GW**2) +$  RRDST155  
 $. (1.14943 - 1.2707E-1*GW + 5.1785E-3*GW**2) * TOF$  RFDST156  
 $FGF = (GR - (-2.7327E-2 + 1.904E-2*GR) * WS +$  RRDST157  
 $. (-6.308077E-4 + 1.94654E-4*GR) * WS**2) * 100.$  RFDST158  
 $GO TO 100$  RRDST159  
26 CCNTINUE RRDST160  
27 CCNTINUE RRDST161  
28 CONTINUE RFDST162  
29  $TOF = (7.83935E-1 + 5.38189E-2*PA) +$  RRDST163  
 $. (1.20408E-2 + 9.888357E-5*PA - 2.32448E-6*PA**2) * T -$  RRDST164  
 $. (9.72E-6 + 1.8278E-6*PA - 2.405E-8*PA**2) * T**2$  RRDST165  
 $GR = (3.18978E+1 - 1.785*GW + 3.602E-2*GW**2) +$  RRDST166  
 $. (-8.8285 + 5.1387E-1*GW - 5.679E-3*GW**2) * TOF +$  RRDST167  
 $. (-1.76441 + 4.82709E-2*GW) * TOF**2$  RRDST168  
 $FGF = (GR - (8.6457E-2 + 1.1414E-2*GR) * WS) * 100.$  RRDST169  
 $GC TO 100$  RFDST170  
30  $TOF = (-2.890514E-1 + 5.8370956E-2*PA) +$  RRDST171  
 $. (4.161561E-2 - 3.518445E-5*PA) * T + (-6.0515E-5 + 3.53095E-6*PA) * T**2$  PRDST172  
 $GF = (-2.684337E+1 + 3.224954*GW) + (-2.0581519 + 3.7024356E-1*GW) * TOF +$  RRDST173  
 $. (-8.861357E-1 + 8.3093188E-2*GW) * TOF**2$  RRDST174  
 $FGF = (GR - (1.3583333E-1 + 9.5833E-3*GR) * WS) * 100.$  RRDST175  
 $GC TO 100$  RRDST176  
31  $TOF = (7.46275E-1 + 1.789924E-2*PA + 1.667729E-4*PA**2) +$  RRDST177  
 $. (6.1017875E-3 + 3.4816947E-4*PA - 1.6406229E-6*PA**2) * T +$  PRDST178  
 $. (1.718525E-4 - 2.621825E-6*PA + 4.184375E-8*PA**2) * T**2$  RRDST179  
 $GR = (-7.2378129E+1 + 3.8485684E+1*GW - 6.565*GW**2 + 3.916E-1*GW**3) +$  RRDST180  
 $. (-5.477E+1 + 2.92E+1*GW - 4.975*GW**2 + 2.906E-1*GW**3) * TOF$  RRDST181  
 $FGF = ((-1.607758 + 1.222176*GR - 5.64375E-3*GR**2) -$  RRDST182  
 $. (.482382E-1 + 2.2260152E-2*GR - 4.7462116E-4*GR**2) * WS) * 100.$  PRDST183  
 $GC TO 100$  RRDST184  
32  $TOF = (1.996 + 1.69E-2*PA + 2.56E-5*PA**2) +$  RRDST185

• (8.64E-3-7.5E-5*PA+1.61F-6*PA**2) *T	FRDST186
GR=(6.26E+1-1.299E+1*GW+6.886E-1*GW**2) +	RRDST187
• (-1.0004E+2+2.0317E+1*GW-9.67E-1*GW**2) *TOF +	RRDST188
• (1.30368E+1-2.689*GW+1.403E-1*GW**2) *TOF**2	PRDST189
FGR=(-3.3E-1+1.047*GR-8.57E-4*GR**2) -	RRDST190
• (4.22E-2+9.47E-3*GR+1.9898E-5*GR) *WS) *100.	RRDST191
GO TO 100	PRDST192
33 TCF=(6.6742857E-1+4.4226786E-2*PA) +	PRDST193
• (1.027143E-2+3.051339E-4*PA) *T+ (1.74994E-4+5.023E-7*PA) *T**2	PRDST194
GR=(-1.37666666E+1+1.679166666*GW)+(-3.55+4.71875E-1*GW)*TOF	PRDST195
FGR=(GR-(1.5166666666E-1+1.008333333E-2*GR)*WS)*100.	RRDST196
GO TO 100	PRDST197
34 CCNTINUE	RRDST198
35 CCNTINUE	PPDST199
36 TCF=(-9.2083337E-1+5.9113889E-2*PA)+(2.291666E-2-2.7778E-5*PA)*T	PRDST200
GR=(3.711176E+1-1.640279E+1*GW+2.22809*GW**2) +	RRDST201
• (-2.09922E+1+8.6991796*GW-8.4586E-1*GW**2)*TOF +	RRDST202
• (2.248949-9.093486E-1*GW+1.061975E-1*GW**2)*TOF**2	PRDST203
FGR=(GR-(4.3358E-2+2.196E-2*GR)*WS+	PRDST204
• (8.79209E-4+8.21219E-5*GR)*WS**2)*100.	RRDST205
GC TO 100	RRDST206
37 TCF=(-6.46E-1+6.7857E-2*PA+2.723E-4*PA**2) +	PRDST207
• (3.69E-2-2.24E-3*PA+3.49E-5*PA**2)*T+	RRDST208
• (1.07E-4+3.85E-5*PA-4.688E-7*PA**2)*T**2	RRDST209
GR=(5.38-1.105*GW+1.14E-1*GW**2) +	PRDST210
• (8.02E-1-2.57E-1*GW+2.4F-2*GW**2)*TOF	PRDST211
FGR=(GR-(1.6E-2+2.44E-2*GR-2.128E-4*GR**2)*WS)*100.	PPDST212
GC TO 100	RRDST213
100 RRDIST=FGR	RRDST214
RETURN	RRDST215
END	PRDST216

FUNCTION SIGY  
(ENTRY: SIGCY)

Purpose:

To compute the horizontal dispersion coefficient in meters, or at entry SIGCY, to compute the virtual distance corresponding to the initial horizontal dispersion.

Input:

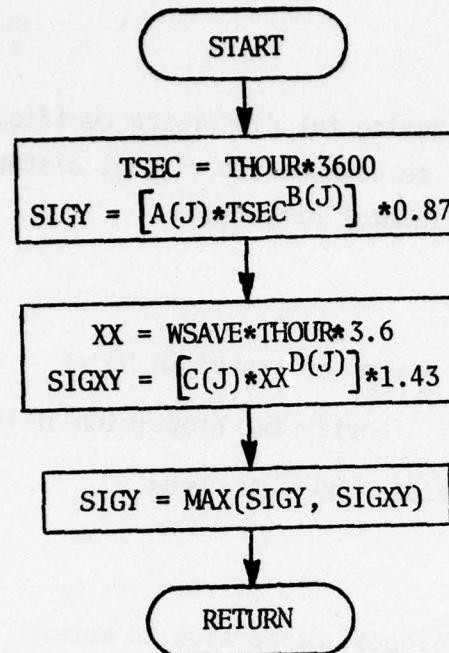
1. Entry SIGY - time of travel in hours
2. Entry SIGCY - horizontal dispersion in meters
3. Stability class and wind speed

Output:

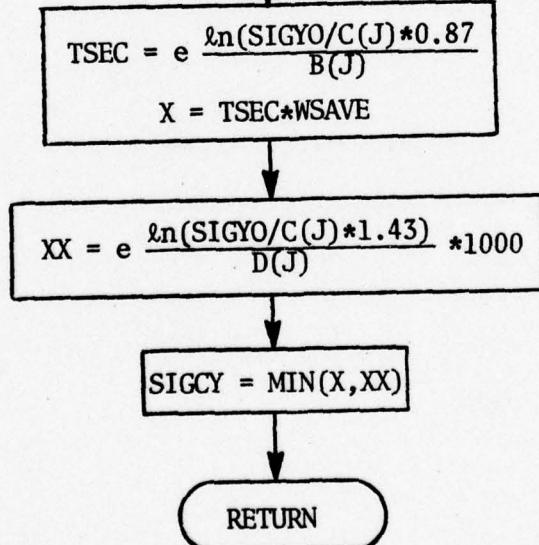
1. SIGY = horizontal dispersion in meters
2. SIGCY = virtual distance in meters

## FUNCTION SIGY

ENTRY: SIGCY



ENTRY: SIGCY



```

FUNCTION SIGY(J,THOUR)
C
C THIS FUNCTION COMPUTES THE HORIZONTAL DISPERSION COEFFICIENT
C IN METERS
C
CCMMON /WDUN/ WSAVE
DIMENSION A(6),B(6),C(6),D(6)
DATA A/2.1511,1.5454,1.0606,.68465,.59366,.59366/
DATA B/.87326,.88261,.89031,.88866,.89138,.89138/
DATA C /212.,155.,100.,68.,50.,34./
DATA D/0.89,0.91,0.92,0.93,0.90,0.93/
C
TSFC=THOUR*3600.
SIGY=(A(J)*TSEC**B(J))*0.87
XX=WSAVE*THOUR*3.6
SIGXY=C(J)*(XX**D(J))*1.43
SIGY=AMAX1(SIGY,SIGXY)
RETURN
ENTRY STGCY(J,SIGY0)
C
C AT THIS ENTRY THE DISTANCE OR TRAVEL TIME CORRESPONDING TO THE
C INPUT VALUE OF THE HORIZONTAL DISPERSION IS CALCULATED AND
C RETURNED AS DISTANCE IN METERS
C
TSEC=EXP(ALOG(SIGY0/(A(J)*0.87))/B(J))
X=TSEC*WSAVF
XX=EXP(ALOG(SIGY0/(C(J)*1.43))/D(J))*1000.
SIGCY=AMIN1(X,XX)
RETURN
END

```

FUNCTION SIGZ  
(ENTRY: SIGCZ)

Purpose:

To compute the vertical dispersion coefficient in meters, or at entry SIGCZ, to compute the virtual distance corresponding to the initial vertical dispersion.

Input:

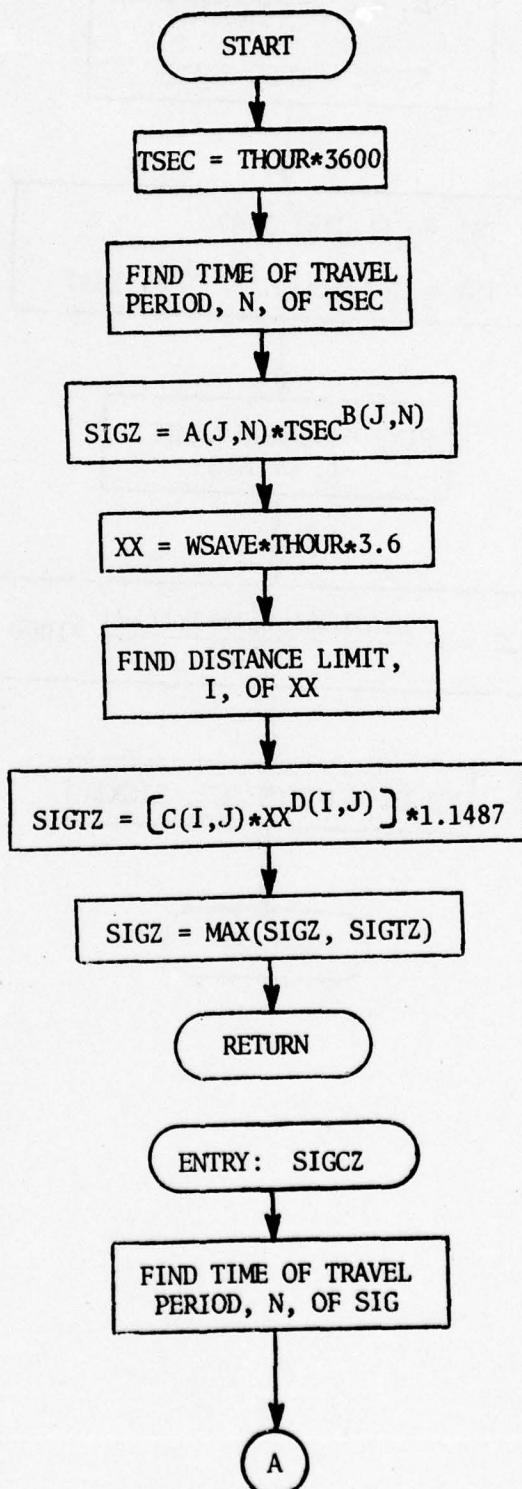
1. Entry SIGZ - time of travel in hours
2. Entry SIGCZ - vertical dispersion in meters
3. Stability class and wind speed

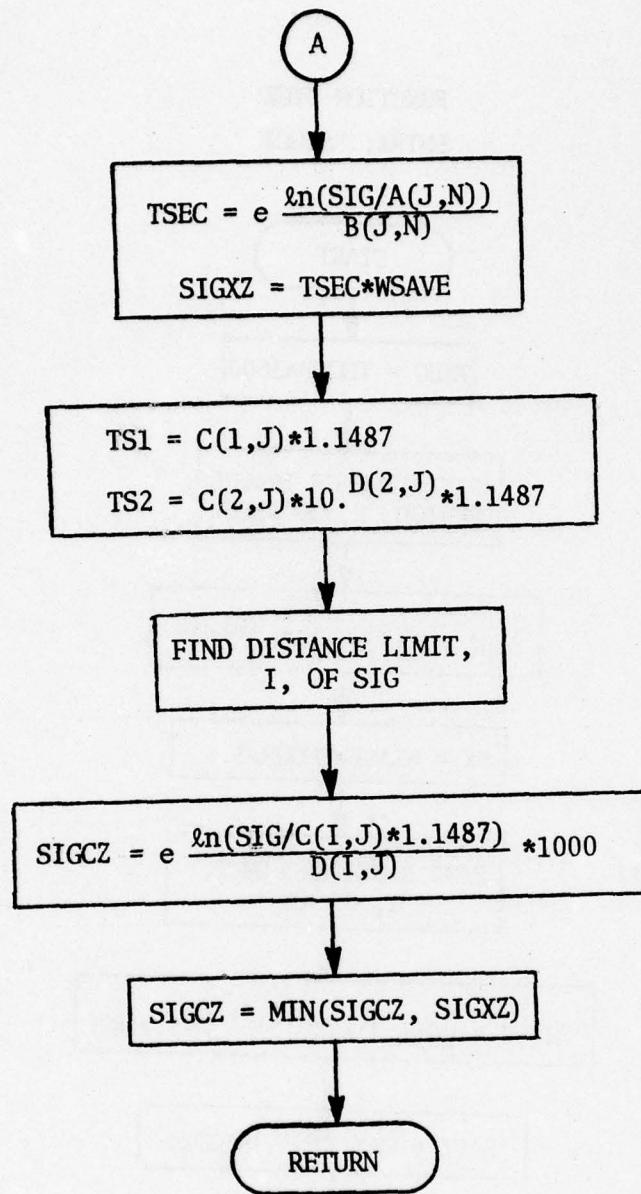
Output:

1. SIGZ = vertical dispersion in meters
2. SIGCZ = virtual distance in meters

FUNCTION SIGZ

ENTRY: SIGCZ





```

FUNCTION SIGZ(J,THOUR)                               SIGZ0000
C                                                 SIGZ0001
C THIS FUNCTION COMPUTES THE VERTICAL DISPERSION COEFFICIENT SIGZ0002
C IN METERS                                         SIGZ0003
C                                                 SIGZ0004
C                                                 SIGZ0005
COMMON /WDUN/ WSAVE                               SIGZ0006
DIMENSION C(3,6),D(3,6),A(6,6),B(6,6),CK(6,6) SIGZ0007
DIMENSION TIME(6)                                 SIGZ0008
DATA TIME/ 300.,1000.,3000.,10000.,30000.,172000./ SIGZ0009
DATA A/ 1.17122,.27668,.41219,.51921,.50963,.47639, SIGZ0010
1      .11062,.39953,.41219,.57145,.76485,.71936, SIGZ0011
2      .01339,.16640,.41219,1.0813,1.9467,2.3901, SIGZ0012
3      .01338,.16640,.41219,2.2830,2.9850,3.8684, SIGZ0013
3      .01338,.16640,.41219,2.3333,5.7990,16.897, SIGZ0014
3      .01338,.16640,.41219,5.6801,14.599,64.577/ SIGZ0015
DATA B/ 1.2098,1.0572,.92365,.84130,.79689,.76308, SIGZ0016
1      1.2864,.99275,.92365,.82449,.72571,.69082, SIGZ0017
2      1.5922,1.1195,.92365,.73217,.59047,.51700, SIGZ0018
3      1.5922,1.1195,.92365,.63883,.53708,.45686, SIGZ0019
4      1.5922,1.1195,.92365,.63646,.46497,.29621, SIGZ0020
5      1.5922,1.1195,.92365,.55016,.37541,.16667/ SIGZ0021
DATA C/ 470.,470.,470.,110.,110.,110.,60.,60.,60.,33.,33.,40., SIGZ0022
. 21.5,21.5,36.,14.,14.,23.5/
DATA D/ 1.67,2.13,2.13,1.,1.09,1.09,0.92,0.92,0.92,0.80,0.61,0.53, SIGZ0023
. 0.70,0.56,0.35,0.78,0.53,0.30/
DATA CK/                                         SIGZ0024
1 170.,    115.,    80.,    63.,    48.,    37.,
2 800.,    380.,   243.25,   170.,   115.,    85.,
3 4600.,   1300.,   671.,   380.,   220.,   150.,
4 31279.,  5002.,  2040.32,  820.,   420.,   260.,
5 179855.2,17111.38,5628.47, 1650.,   700.,   358.,
6 2900444.,120872.5,28241.86,4312.55,1348.32,481.58/ SIGZ0025
C
C TSEC=THOUR*3600.                               SIGZ0026
DO 10 N=1,6                                     SIGZ0027
IF(TSEC.LE.TIME(N))GO TO 20                     SIGZ0028
10 CONTINUE                                     SIGZ0029
N=6
C
C TIME OF TRAVEL SHOULD BE LESS THAN 172000 SEC. OR APPROX. 2 DAYS SIGZ0030
C
20 CONTINUE                                     SIGZ0031
SIGZ=(A(J,N)*TSEC**B(J,N))                     SIGZ0032
XX=WSAVE*THOUR*3.6                               SIGZ0033
I=1
TF(XX.GT.1.) T=2                               SIGZ0034
TF(XX.GT.10.) T=3                               SIGZ0035
C
C CONVERTS FROM A 10 TO 20 MIN. SAMPLING TIME SIGZ0036
C 1.1487 = 2**.2, THE 1/5 POWER LAW ONLY APPLIES UP TO 20 MIN. SIGZ0037
C SAMPLING TIMES                                   SIGZ0038
C
SIGTZ=(C(I,J)*XX**D(I,J))*1.1487             SIGZ0039
SIGZ=AMAX1(SIGZ,SIGTZ)                         SIGZ0040
RETURN                                           SIGZ0041
ENTRY SIGCZ(J,SIG)                            SIGZ0042
C
C AT THIS ENTRY THE DISTANCE OR TRAVEL TIME CORRESPONDING TO THE SIGZ0043
C INPUT VALUE OF THE VERTICAL DISPERSION IS CALCULATED AND SIGZ0044
C RETURNED AS DISTANCE IN METERS                 SIGZ0045
C
DO 110 N=1,6                                     SIGZ0046

```

```
110 IF (SIG.LE.CK(J,N)) GO TO 120           SIGZ0062
CONTINUE                                     SIGZ0063
N=6                                         SIGZ0064
120 CONTINUE                                     SIGZ0065
TSEC=EXP(ALOG(SIG/A(J,N))/B(J,N))          SIGZ0066
SIGXZ=TSEC*SAVE                            SIGZ0067
TS1=C(1,J)*1.1487                          SIGZ0068
TS2=C(2,J)*10.**D(2,J)*1.1487            SIGZ0069
I=3                                         SIGZ0070
IF (SIG.LT.TS2) I=2                         SIGZ0071
IF (SIG.LT.TS1) I=1                         SIGZ0072
SIGCZ=EXP(ALOG(SIG/(C(I,J)*1.1487))/D(I,J))*1000. SIGZ0073
SIGCZ=AMIN1(SIGCZ,SIGXZ)                   SIGZ0074
RFTURN                                     SIGZ0075
END                                         SIGZ0076
```

## SUBROUTINE SOURCE

### Purpose:

To position the master source tape to read the airbase and environ source inventory data and to call the subroutines which compute the emission rates in micrograms per second at the airbase and environ sources.

### Input:

JFLAG, a parameter to indicate whether the diurnal distribution used is input, default or the same as previous hour.

### Output:

A statement indicating the diurnal distribution used.

### Subroutines

#### Called:

ABPTAR, ABARAR, ABLNAR, ENARAY

SUBROUTINE SOURCE	SOURC001
C	SOURC002
C THIS ROUTINE SERVES AS A DRIVER TO CALL SUBROUTINES	SOURC003
C WHICH COMPUTE THE EMISSION RATES IN MICROGRAMS	SOURC004
C PER SECOND AT THE AIRBASE AND ENVIRON SOURCES	SOURC005
C	SOURC006
COMMON / DEFAUT / ITAPE	SOURC007
COMMON / PERIOD / IMONTH, NODAYS, IDAY, IHR1, IHR2, IFLAG, JFLAG, IONCE	SOURC008
COMMON / JUNK / DAYS, LSRCE, NSRCE, SORCE(17,300), SORG(10,200)	SOURC009
, LCC1, LOC2, NGEOM, IPT	SOURC010
DIMENSION NAME(2)	SOURC011
DATA NAME / 4H1/12,4H1   /	SOURC012
IF (ICNCE.EQ.0) GO TO 30	SOURC013
IEND=1	SOURC014
IST=1	SOURC015
GC TO 40	SOURC016
30 IEND=0	SOURC017
ICNCE=1	SOURC018
IST=0	SOURC019
40 CCNTINUE	SOURC020
DAYS=NODAYS	SOURC021
IF (IST.EQ.1) GO TO 3	SOURC022
1 FORMAT (I4)	SOURC023
READ 1, JFLAG	SOURC024
IF (JFLAG) 8,7,3	SOURC025
7 EFINT 5	SOURC026
5 FORMAT(32H0INPUT DIURNAL DISTRIBUTION USED)	SOURC027
GC TC 4	SOURC028
8 I=1	SOURC029
IF (NCDAYS.EQ.365) I=2	SOURC030
PRINT 9,NAME(I)	SOURC031
9 FORMAT(34H0DEFAULT DIURNAL DISTRIBUTION USED/5X,12HHOUR = 1/24,5X,	SOURC032
.10HDAY = 1/7,5X,8HMONTH = A4,1H,5X,12HUNIFRC = 0.1)	SOURC033
GC TC 4	SOURC034
3 PRINT 6	SOURC035
6 FORMAT(39H0DIURNAL DISTRIBUTION SAME AS LAST HOUR)	SOURC036
GC TC 10	SOURC037
4 IF(IEND.EQ.0) GO TO 12	SOURC038
11 READ (ITAPE,END=12)	SOURC039
GC TC 11	SOURC040
12 IEND=1	SOURC041
CALL ABFTAR	SOURC042
CALL AEAKAR	SOURC043
CALL ABLNAR	SOURC044
CALL ENARAY	SOURC045
10 REWIND ITAPE	SOURC046
RETURN	SOURC047
END	SOURC048

## SUBROUTINE STPOL1

### Purpose:

To calculate pollutant concentrations from point and area sources.

### Input:

1. Location and conditions at point or area source
2. Location of receptors
3. Meteorological conditions

### Output:

Concentration of pollutants at each receptor

### Procedure:

1. For area sources determine average diameter, effective stack height, and initial values of horizontal and vertical dispersion. Also consider effects of downwash on these.
2. For point sources determine plume rise by calling PLRISE.
3. Consider effects of wind speed at height of source.
4. Calculate crosswind and downwind components for the source.
5. Calculate time required for plume to travel from virtual point source to actual location of true source by PSEUDO routine.
6. For each receptor, calculate crosswind and downwind components.
7. Consider the relative location of receptor with respect to source and, if necessary, calculate coupling coefficient using TRAN routine.
8. For each pollutant, add in concentration determined.

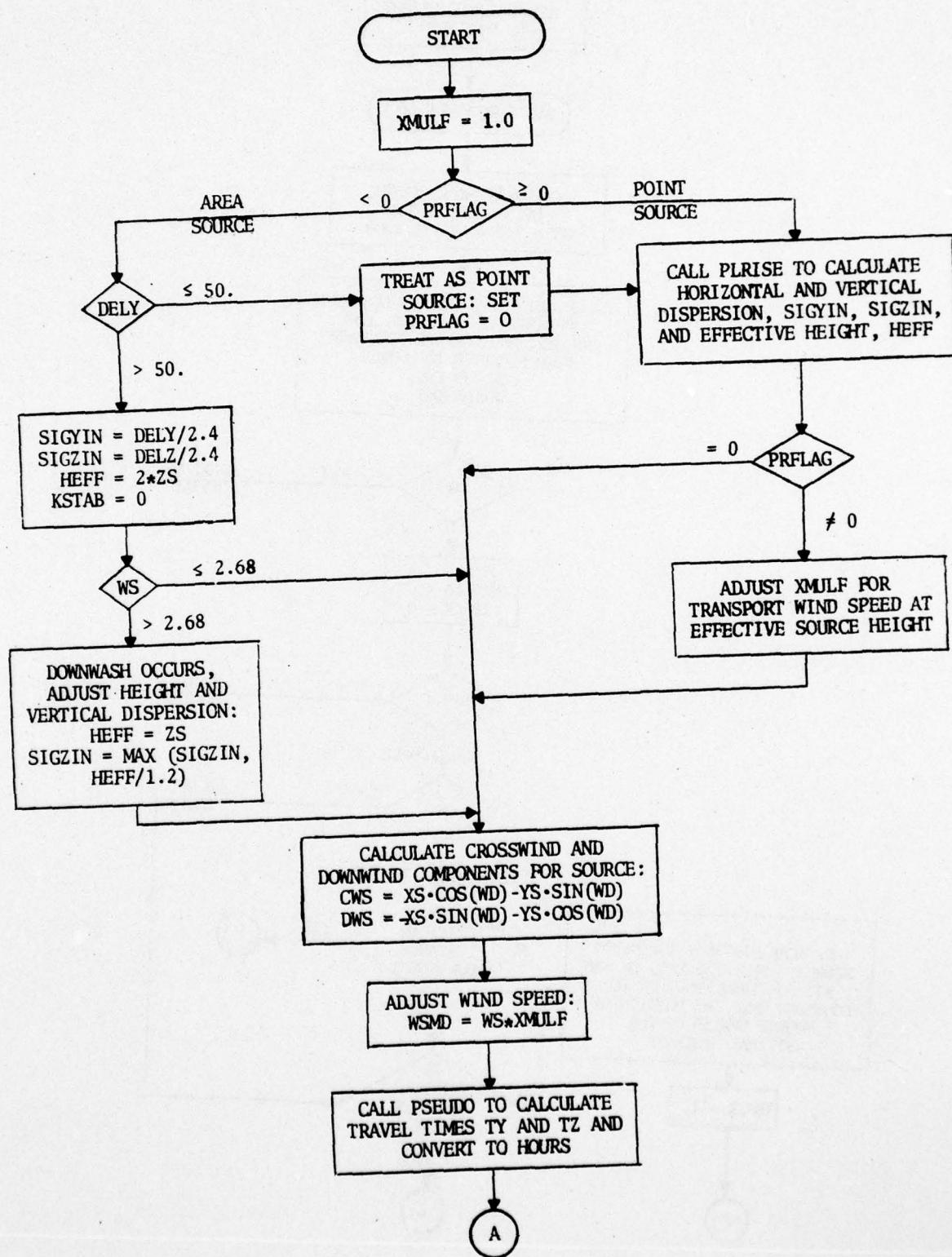
### Subroutines Called:

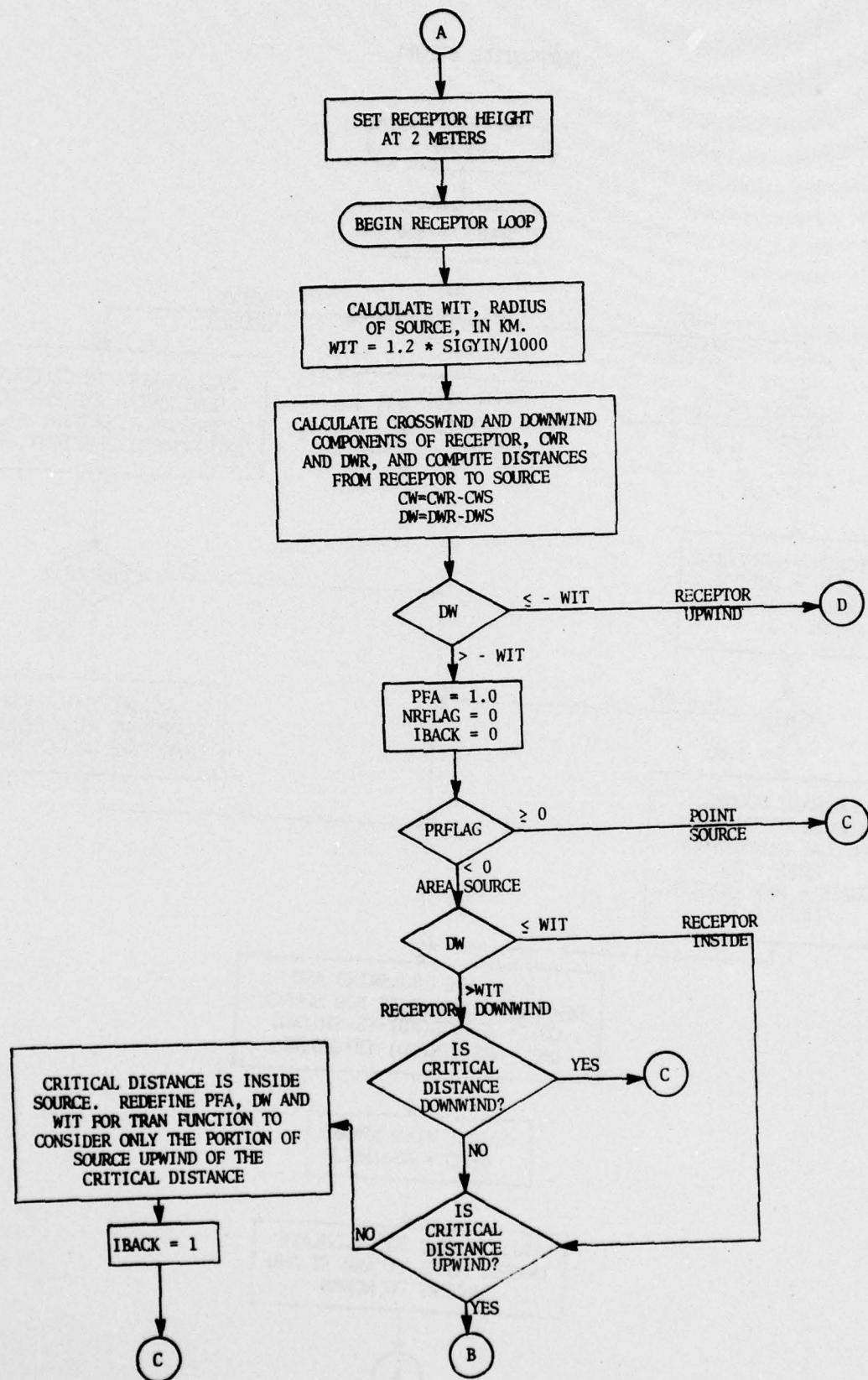
PLRISE, PSEUDO

### Function Called:

TRAN

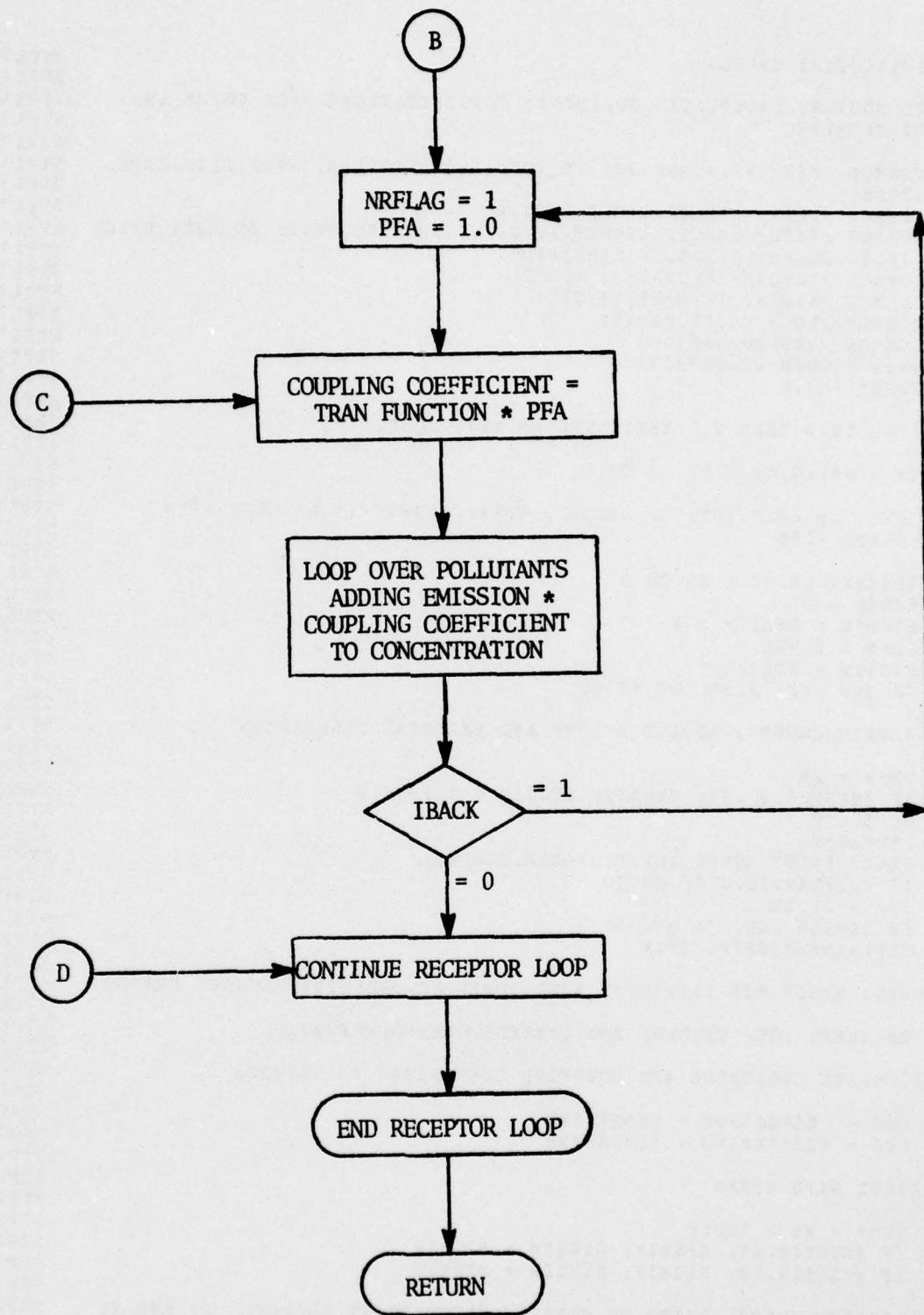
SUBROUTINE STPOL1





CRITICAL DISTANCE IS INSIDE SOURCE. REDEFINE PFA, DW AND WIT FOR TRAN FUNCTION TO CONSIDER ONLY THE PORTION OF SOURCE UPWIND OF THE CRITICAL DISTANCE

IBACK = 1



```

SUBROUTINE STPOL1 STPL1000
C THIS ROUTINE CALCULATES POLLUTANT CONCENTRATIONS FROM POINT AND STPL1001
C AREA SOURCES STPL1002
C
COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP, STPL1003
C . TENV STPL1004
COMMON /BCPT/ NRECEP,RECEP(2,312) STPL1005
COMMON /INFO/ IRECEP,IWNDIR,ITYPE ,HTAERO,XS,YS,ZS,DELY,DELZ, STPL1006
C . TS,VS,DS,HB,PRFLAG,EMIS(8),NPOL STPL1007
COMMON /AIRQAL/ RECDAT(3, 6,312) STPL1008
COMMON /XTRAN/ XL,WSMD,TY,TZ STPL1009
COMMON /LOC/ DW,CW,ZR,WIT STPL1010
COMMON /WNDPRO/ XP(6) STPL1011
DATA SIGMIN /2.083333/ STPL1012
XMULF = 1.0 STPL1013
C PRFLAG LESS THAN 0.0 INDICATES AN AREA SOURCE STPL1014
C
IF (PRFLAG.GE.0.0) GO TO 5 STPL1015
C
IF DELY IS LESS THAN 50 METERS, TREAT SOURCE AS A POINT WITH STPL1016
C NO FLUME RISE STPL1017
C
IF(DELY.LE.50.) GO TO 4 STPL1018
KSTAB = 0 STPL1019
SIGYIN = DELY / 2.4 STPL1020
HEFF = 2.*ZS STPL1021
SIGZIN = DELZ/2.4 STPL1022
IF (WS .LE. 2.68) GO TO 10 STPL1023
C
DOWNWASH OCCURS, ADJUST HEIGHT AND VERTICAL DISPERSION STPL1024
C
HEFF = ZS STPL1025
IF (HEFF/1.2 .GT. SIGZIN) SIGZIN = HEFF/1.2 STPL1026
GC TO 10 STPL1027
4 PRFLAG=0. STPL1028
5 CALL FLRISE (HEFF,KSTAB,SIGZIN,SIGYIN) STPL1029
IF (PRFLAG.EQ.0.0) GO TO 10 STPL1030
JJJ = JSTAB STPL1031
IF (KSTAB .GE. 1) JJJ=5 STPL1032
HEFL=AMIN1(HEFF,305.)
C
ADJUST XMULF FOR TRANSPORT WIND SPEED AT EFFECTIVE SOURCE HEIGHT STPL1033
C
IF (HEFL .GT. HTAERO) XMULF=(HEFL/HTAERO)**XP(JJJ) STPL1034
C
CALCULATE CROSSWIND AND DOWNWIND COMPONENTS FOR SOURCE STPL1035
C
10 CWS = XS*COSEWD - YS*SINEWD STPL1036
DWS = -XS*SINEWD - YS*COSEWD STPL1037
C
ADJUST WIND SPEED STPL1038
C
WSMD = WS * XMULF STPL1039
IF (SIGYIN.LT. SIGMIN) SIGYIN = SIGMIN STPL1040
IF (SIGZIN.LT. SIGMIN) SIGZIN = SIGMIN STPL1041
C
CALCULATE TRAVEL TIMES TO PSEUDO UPWIND POINT SOURCE. TY AND TZ STPL1042
C ARE RETURNED AS DISTANCES IN METERS AND CONVERTED TO EQUIVALENT STPL1043
C TIMES IN HOURS STPL1044
C

```

```

CALL FSEUDO(SIGYIN,WSMD,SIGZIN,TY,TZ) STPL1062
TY=TY/WSMD/3600. STPL1063
TZ=TZ/WSMD/3600. STPL1064
ZR = 2. STPL1065
C STPL1066
C BEGIN RECEPTOR LOOP STPL1067
C STPL1068
C DC 2C NR=1,NRECEP STPL1069
C STPL1070
C WIT IS RADIUS OF SOURCE IN KILOMETERS STPL1071
C STPL1072
C WIT = 1.2 * SIGYIN / 1000. STPL1073
C STPL1074
C CALCULATE CROSSWIND AND DOWNWIND COMPONENTS OF RECEPTOR AND STPL1075
C COMPUTE DISTANCES IN KILOMETERS FROM RECEPTOR TO SOURCE STPL1076
C STPL1077
C CWR = RECEP(1,NR)*COSEWD - RECEP(2,NR)*SINEWD STPL1078
C CW = CWR - CWS STPL1079
C DWR = -RECEP(1,NR) * SINEWD - RECEP(2,NR) * COSEWD STPL1080
C DW = DWR - DWS STPL1081
C STPL1082
C PFA IS FRACTION OF TOTAL AREA SOURCE BEING TREATED STPL1083
C STPL1084
C IF (DW.LE.-WIT) GO TO 20 STPL1085
C STPL1086
C IS RECEPTOR UPWIND OF SOURCE? STPL1087
C STPL1088
C PFA = 1.0 STPL1089
C NRFLAG=0 STPL1090
C IBACK=0 STPL1091
C IF (PRFLAG.GE.0.0) GO TO 16 STPL1092
C STPL1093
C IS RECEPTOR INSIDE SOURCE? STPL1094
C STPL1095
C IF(DW.LE.WIT) GO TO 15 STPL1096
C STPL1097
C IS CRITICAL DISTANCE DOWNWIND OF DOWNWIND EDGE OF SOURCE? STPL1098
C STPL1099
C IF((DWR-XL).GE.(DWS+WIT)) GO TO 16 STPL1100
C STPL1101
C IS CRITICAL DISTANCE UPWIND OF UPWIND EDGE OF SOURCE? STPL1102
C STPL1103
C 15 IF((DWR-XL).LT.(DWS-WIT)) GO TO 17 STPL1104
C STPL1105
C CRITICAL DISTANCE IS INSIDE SOURCE. REDEFINE PFA, DW AND WIT STPL1106
C FCR TRAN FUNCTION TO CONSIDER ONLY THE PORTION OF SOURCE STPL1107
C UPWIND OF THE CRITICAL DISTANCE STPL1108
C STPL1109
C PFA=(DW-XL+WIT)/(2.*WIT) STPL1110
C DWSS=((DWS-WIT)+(DWR-XL))/2. STPL1111
C DW=DWR-DWSS STPL1112
C WIT=DW-XL STPL1113
C IBACK=1 STPL1114
C GO TO 16 STPL1115
C 17 NRFLAG=1 STPL1116
C PFA=1. STPL1117
C STPL1118
C CALL TRAN FUNCTION TO DETERMINE COUPLING COEFFICIENTS STPL1119
C STPL1120
C 16 CUPCOE = TRAN (KSTAB,HEFF,NRFLAG,IBACK) * PFA STPL1121
C STPL1122
C ADD EMISSIONS TIMES COUPLING COEFFICIENT TO CONCENTRATIONS STPL1123

```

C AT ALL RECEPATORS STPL1124  
C STPL1125  
DO 18 IPOL = 1,NPOL STPL1126  
18 RECDAT(ITYPE,IPOL,NR) = RECDAT(ITYPE,IPOL,NR) + EMIS(IPOL) \*CUPCOESTPL1127  
IF(IBACK.EQ.1) GO TO 17 STPL1128  
20 CONTINUE STPL1129  
RETURN STPL1130  
END STPL1131

## SUBROUTINE STPOL2

### Purpose:

To prepare the data required by the line source model and, for each receptor, to call the model and then add the pollutant concentrations calculated to the accumulated totals at that receptor.

### Input:

1. Source parameters for the current line.
2. Wind speed and direction, and lid height.

### Output:

Accumulated pollutant concentrations at all receptors.

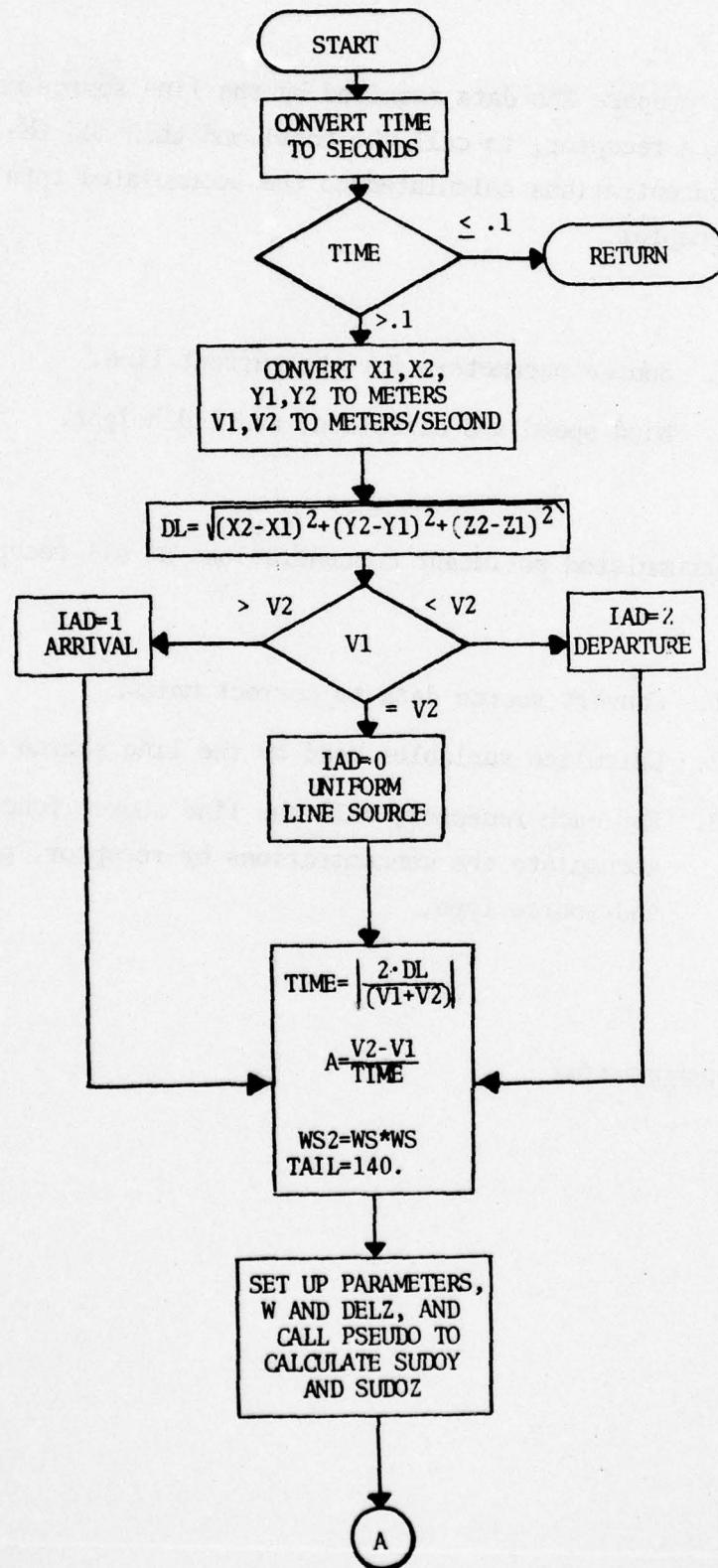
### Procedure:

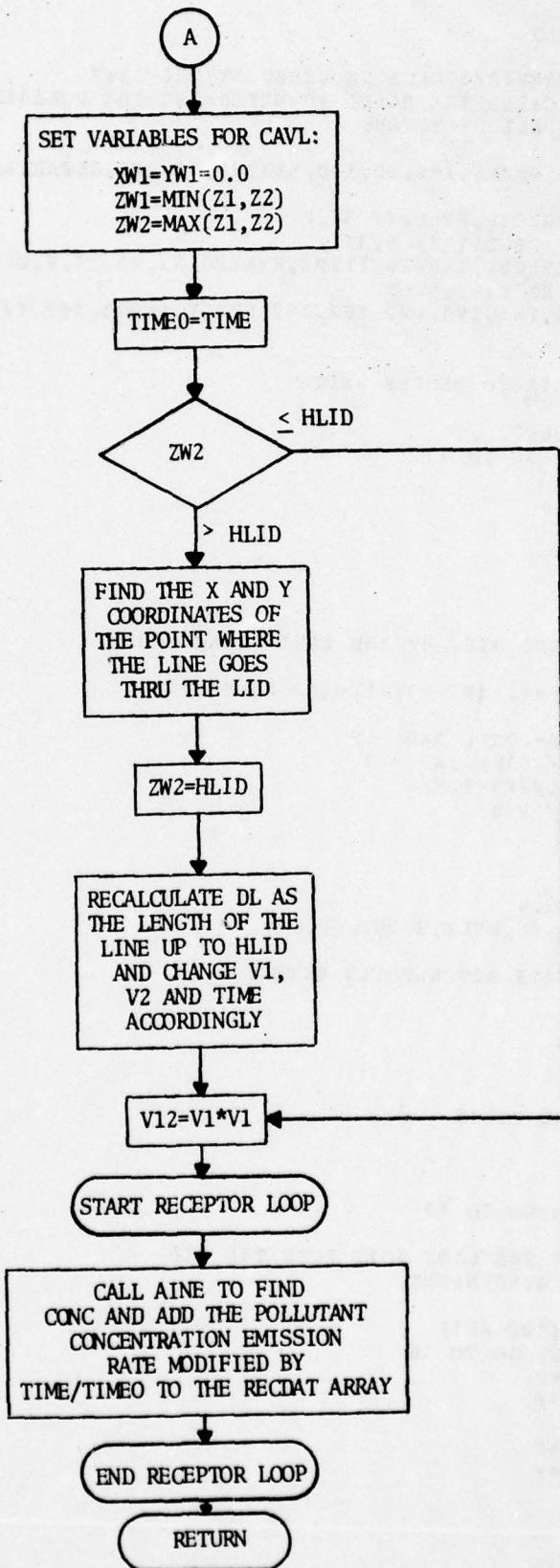
1. Convert source data to correct units.
2. Calculate variables used by the line source model.
3. For each receptor, call the line source function and accumulate the concentrations by receptor, pollutant, and source type.

### Subroutines Called:

PSEUDO,AINE

SUBROUTINE STPOL2





AD-A046 348

ARGONNE NATIONAL LAB ILL  
AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-T--ETC(U)  
APR 77 D J BINGAMAN

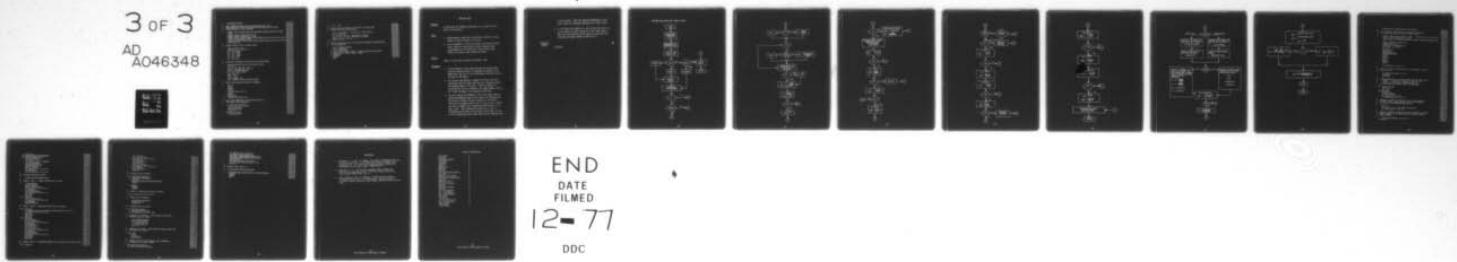
F/G 13/2

UNCLASSIFIED

CEEDO-TR-76-34

NL

3 OF 3  
AD  
A046348



END  
DATE  
FILED  
12-77  
DDC

```

SUBROUTINE STPOL2 STPL2000
C
C THIS SUBROUTINE PREPARES DATA REQUIRED BY THE LINE STPL2001
C SOURCE MODEL AND CALLS THE MODEL TO DETERMINE THE POLLUTANT STPL2002
C CONCENTRATIONS AT ALL RECEPTORS STPL2003
C STPL2004
C STPL2005
COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP, STPL2006
. TEMP STPL2007
COMMON /RCPT/ NRECEP,RECEP(2,312) STPL2008
COMMON /AIRQAL/ RECDAT(3, 6,312) STPL2009
COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,STPL2010
. V1,V2,DL,TIME,EMIS(6),NPOL STPL2011
COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,STPL2012
. WS2,WSC,RR,SP STPL2013
C
C CONVERT SOURCE DATA TO PROPER UNITS STPL2014
C STPL2015
C STPL2016
TIME = TIME * 3600. STPL2017
IF(TIME.LE.0.1) GO TO 11 STPL2018
X1 = X1 * 1000. STPL2019
Y1 = Y1 * 1000. STPL2020
X2 = X2 * 1000. STPL2021
Y2 = Y2 * 1000. STPL2022
V1 = V1 / 3.6 STPL2023
V2 = V2 / 3.6 STPL2024
C
C CALCULATE VARIABLES USED BY THE LINE SOURCE MODEL STPL2025
C STPL2026
C STPL2027
DL=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2) STPL2028
IAD = 0 STPL2029
IF (V1 .LT. (V2-.01)) IAD = 2 STPL2030
IF (V1 .GT. (V2+.01)) IAD = 1 STPL2031
TIME = ABS(2*DL/(V1+V2)) STPL2032
A = (V2 - V1) / TIME STPL2033
WS2 = WS * WS STPL2034
TAIL = 140. STPL2035
W = W / 2.4 STPL2036
DELZ = DELZ / 2.4 STPL2037
CALL PSEUDO (W,WS,DELZ,SUDOY,SUDOZ) STPL2038
C
C SUDOY AND SUDOZ ARE RETURNED IN METERS STPL2039
C STPL2040
C STPL2041
XW1=0. STPL2042
YW1=0. STPL2043
ZW1=Z1 STPL2044
ZW2=Z2 STPL2045
IF(Z1.LE.Z2) GO TO 15 STPL2046
Z1=Z2 STPL2047
Z2=Z1 STPL2048
15 TIME0=TIME STPL2049
IF(ZW2.LE.HLID) GO TO 18 STPL2050
C
C FIND POINT WHERE THE LINE GOES THRU THE LID STPL2051
C AND CHANGE THE COORDINATES STPL2052
C STPL2053
C STPL2054
F=(HLID-ZW1)/(ZW2-ZW1) STPL2055
IF (Z1 .GT. Z2) GO TO 16 STPL2056
X2=X1+(X2-X1)*F STPL2057
Y2=Y1+(Y2-Y1)*F STPL2058
GC TO 17 STPL2059
16 X1=X2+(X1-X2)*F STPL2060
Y1=Y2+(Y1-Y2)*F STPL2061

```

```

17 ZW2 = HLID STPL2062
C STPL2063
C RECALCULATE THE LENGTH OF THE LINE UP TO HLID AND STPL2064
C CHANGE VELOCITIES ACCORDINGLY STPL2065
C STPL2066
C STPL2067
DLSQ = (X1-X2)**2 + (Y1-Y2)**2 + (ZW1-ZW2)**2 STPL2068
DL = SQRT(DLSQ) STPL2069
IF (Z2 .GT. Z1) V2 = SQRT(V1*V1+2.*A*DL) STPL2070
IF (Z2 .LT. Z1) V1 = SQRT(V2*V2-2.*A*DL) STPL2071
TIME = 2* DL / (V1+V2) STPL2072
18 V12 = V1 * V1 STPL2073
C STPL2074
C CALL THE LINE FUNCTION TO DETERMINE POLLUTANT CONCENTRATIONS STPL2075
C AT ALL RECEPTORS STPL2076
C STPL2077
DO 10 IRECEP=1,NRECEP CONC = AINE(WD) STPL2078
DO 10 IPOL=1,NPOL STPL2079
10 RECDAT (ITYPE,IPOL,IRECEP) = RECDAT(ITYPE,IPOL,IRECEP) +
- EMIS(IPOL) * CONC * TIME / TIME0 STPL2080
STPL2081
11 CCONTINUE STPL2082
RETURN. STPL2083
END STPL2084

```

## FUNCTION TRAN

### Purpose:

To calculate the coupling coefficient at a receptor due to a point or area source.

### Input:

1. Meteorological conditions: wind speed, stability, mixing height, critical distance for mixing.
2. Source parameters: initial horizontal and vertical dispersion; effective stack height; pseudo transport times corresponding to the dispersions; plume height flag, KSTAB; area source flags, NRFLAG and IBACK.

### Output:

Point or area source coupling coefficient, TRAN.

### Procedure:

1. If the effective stack height exceeds the mixing height, then the stability index is reassigned according to the KSTAB flag, the lid is set at 3050 meters and the critical distance at 100 meters.
2. For sources with NRFLAG=0, compute the travel time for z dispersion from the center and that for y dispersion from the downwind edge of the source. Then the effects of ground and sky lid are treated by the image method, with up to 6 terms included in the coupling coefficient.
3. For area sources with NRFLAG=1, the travel times from the upwind and downwind edges of the source are determined on the basis of receptor location relative to the source. These plus the pseudo travel time, TZ, due to the z spread are used to compute the z-dispersion coefficients  $\sigma_z$  (T1) and  $\sigma_z$  (T2). The y-dispersion coefficient  $\sigma_y$  (TY) is determined on the basis of the pseudo travel time, TY, due to the y-spread plus the travel time from the downwind edge

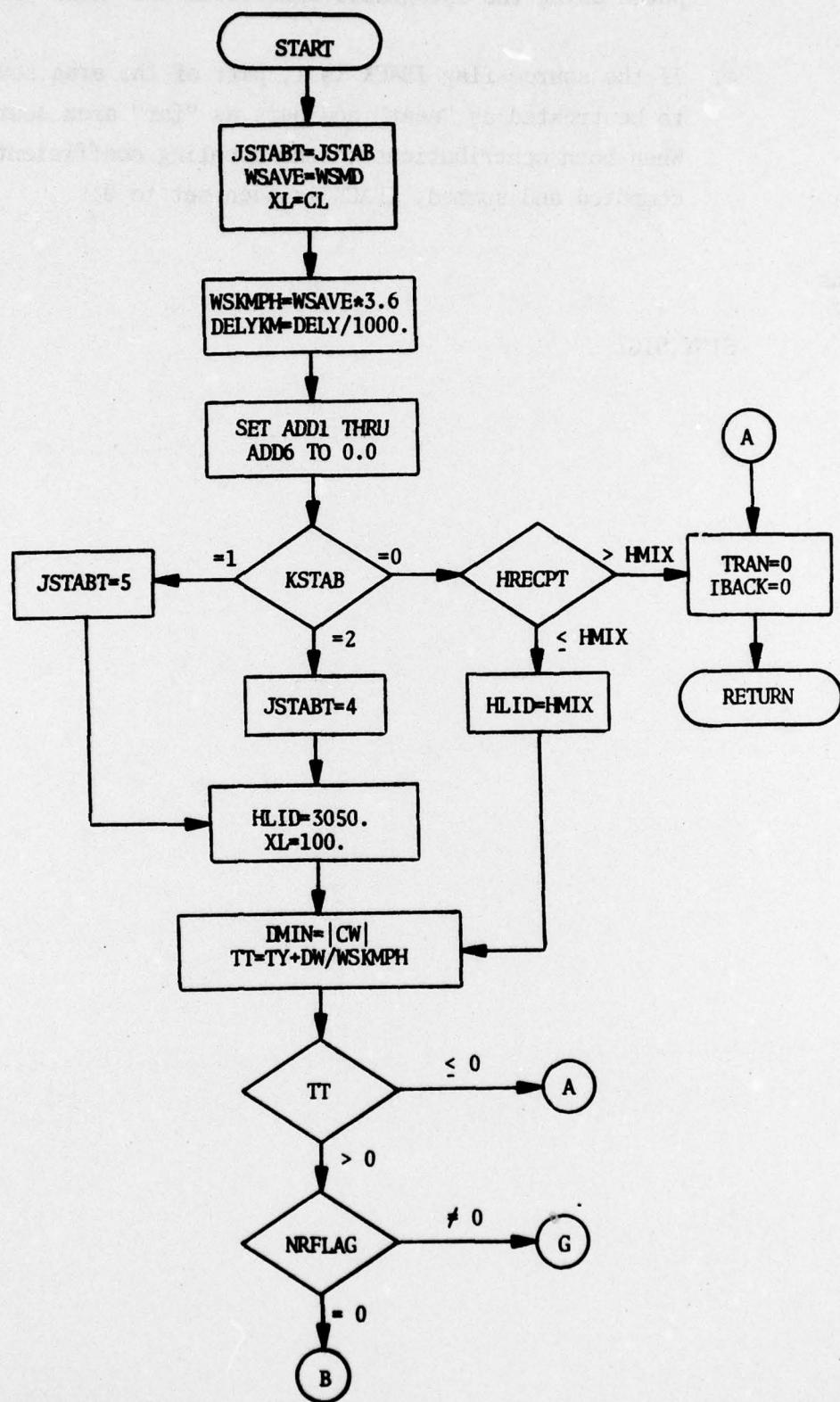
to the receptor. Then the coupling coefficient is computed using the integrated expression for "near" source.

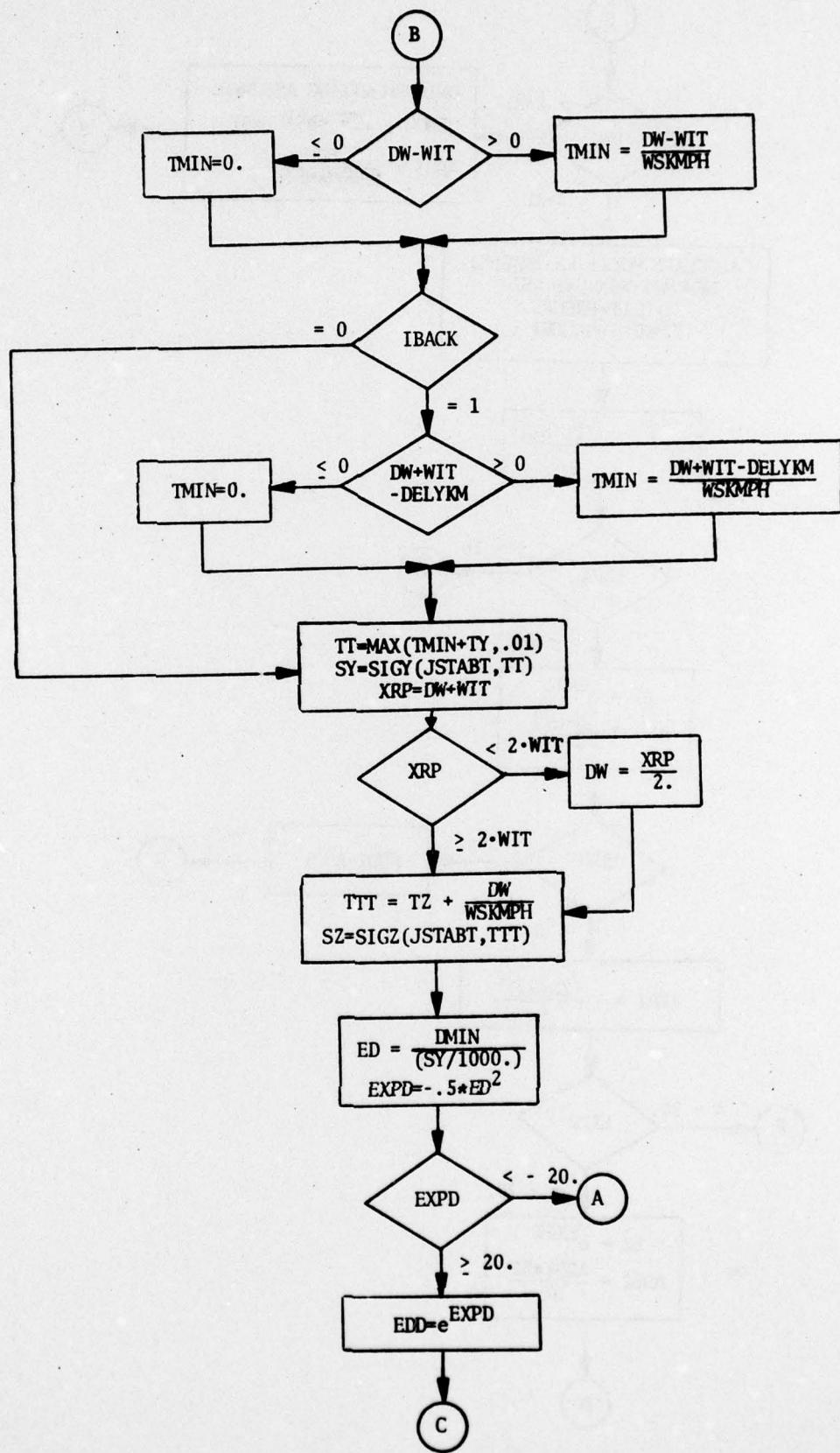
4. If the source flag IBACK is 1, part of the area source is to be treated as "near" and part as "far" area sources. When both contributions to the coupling coefficient are computed and summed, IBACK is then set to 0.

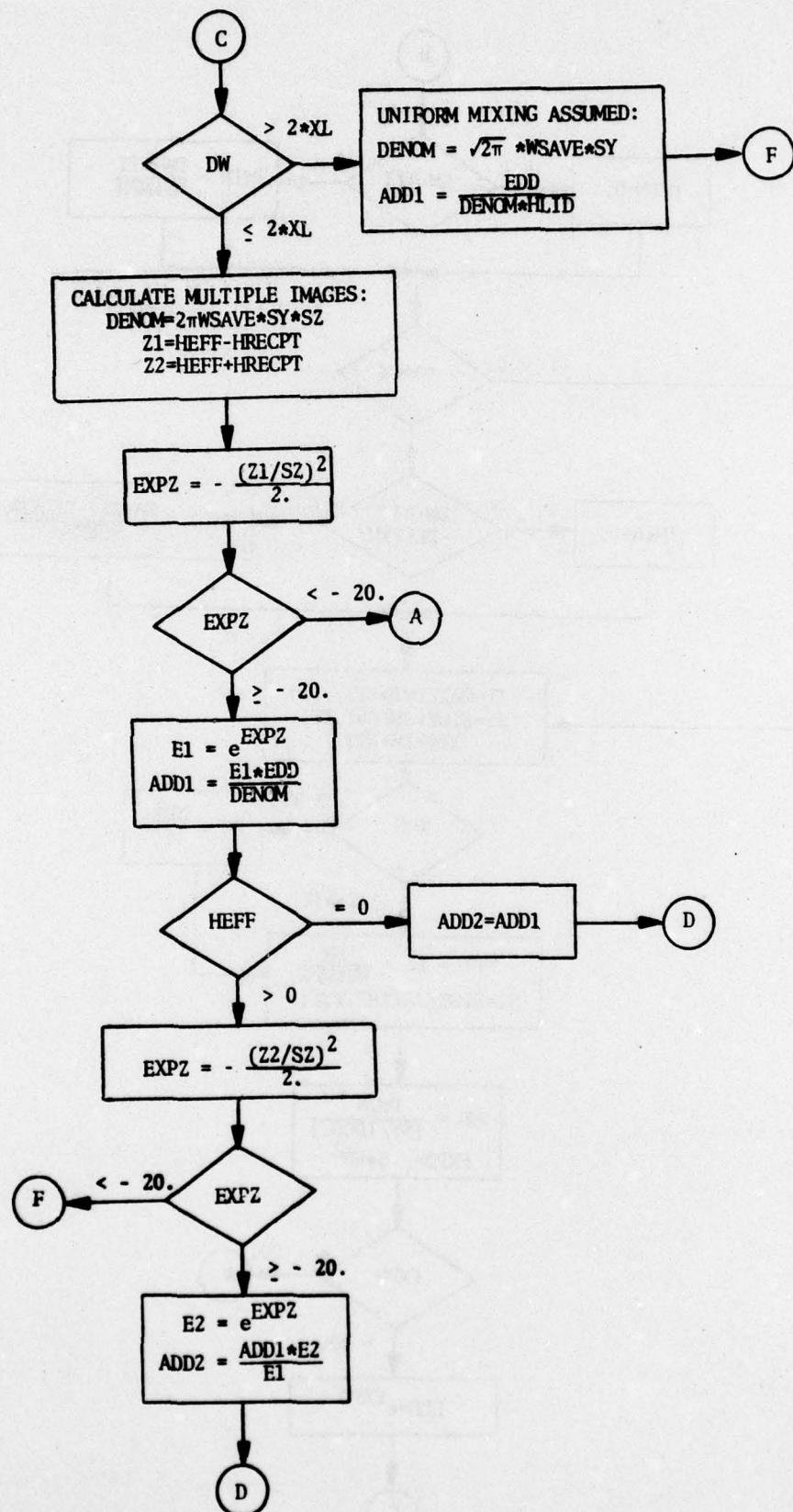
Functions  
Called:

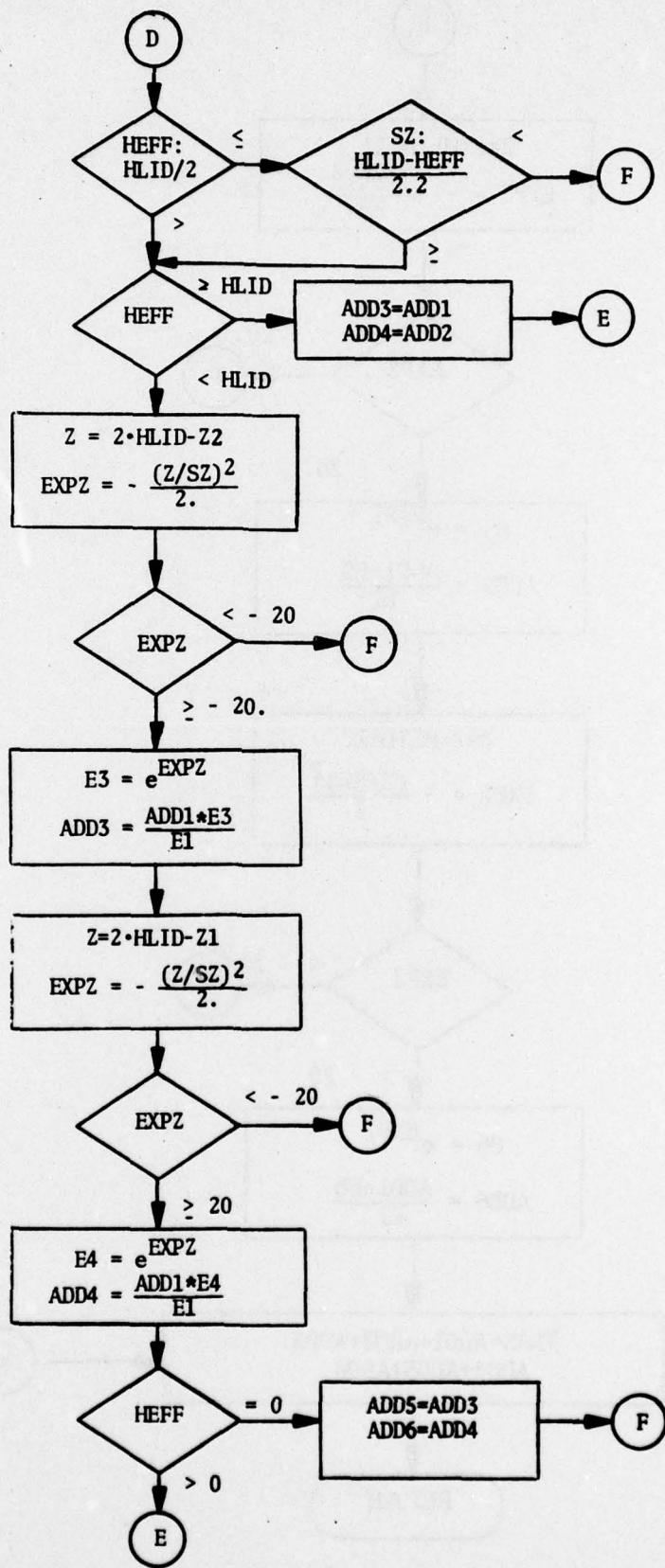
SIGY,SIGZ

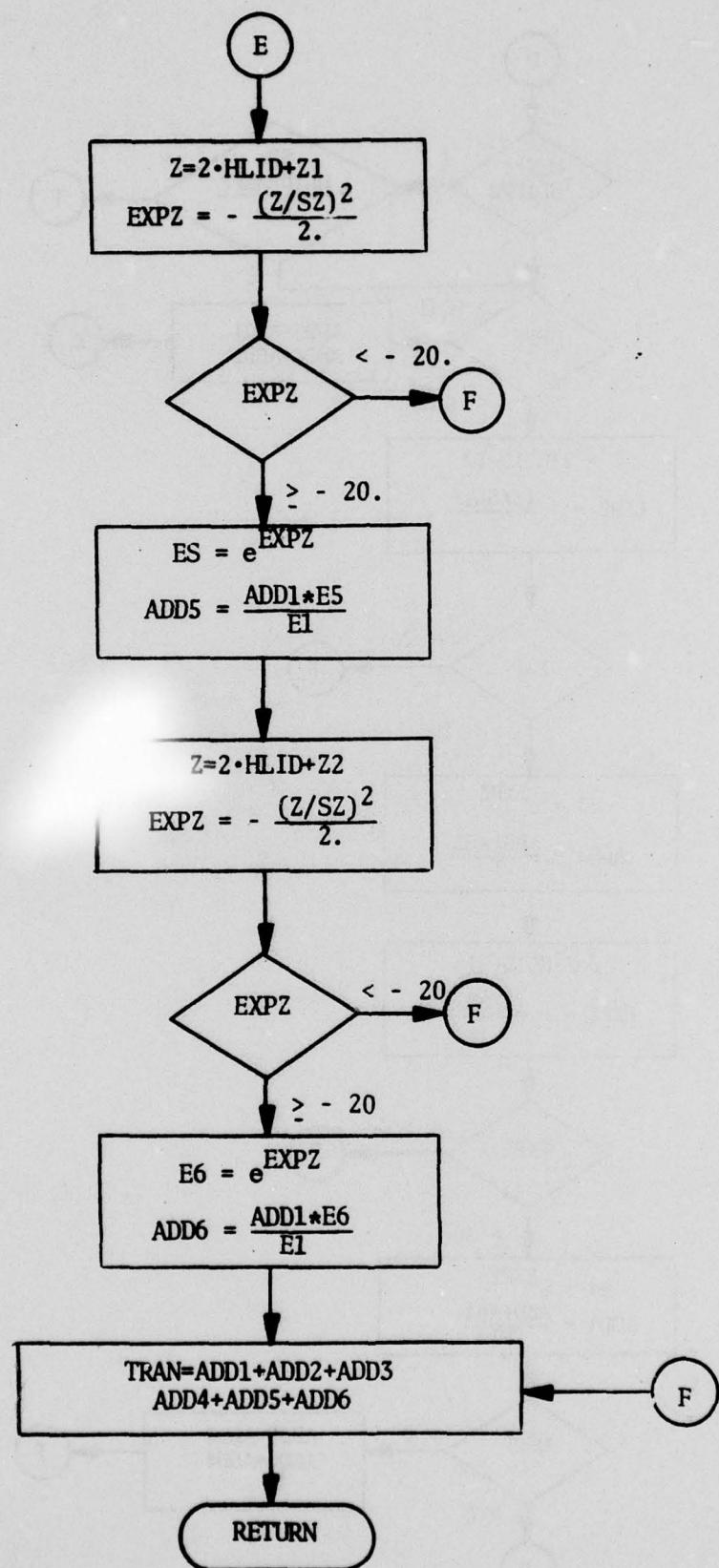
FUNCTION TRAN (KSTAB, HEFF, NRFLAG, IBACK)

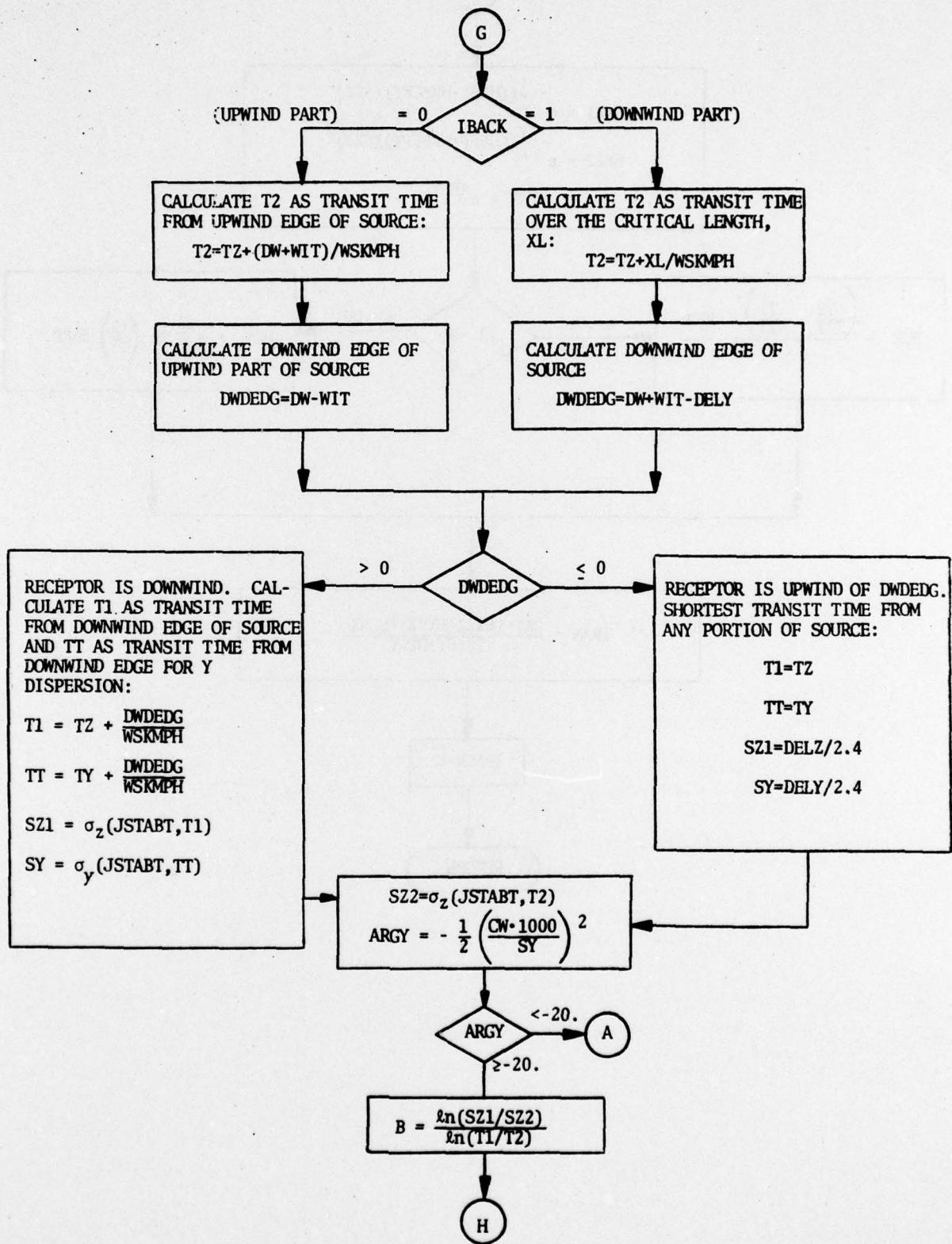


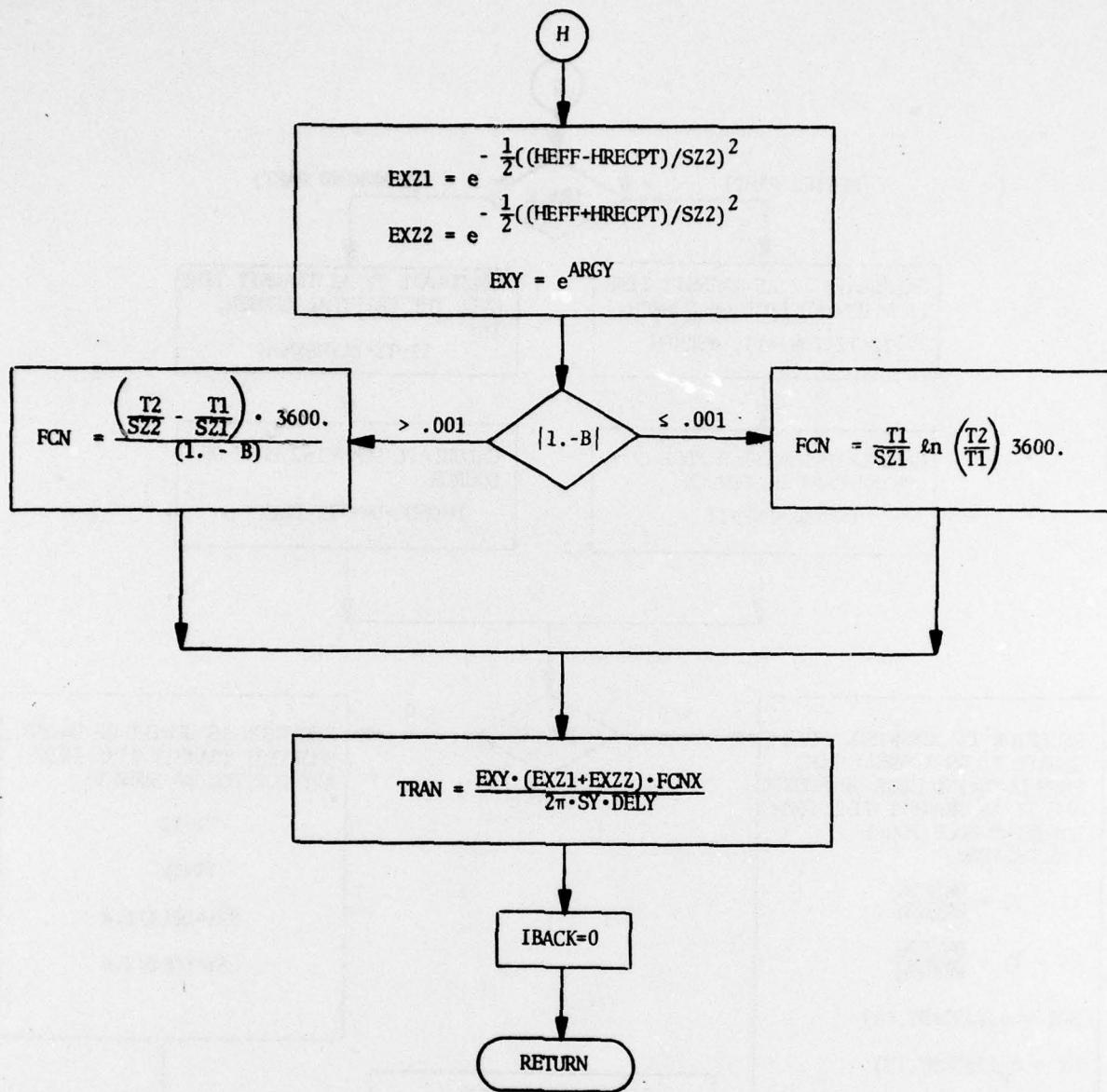












```

FUNCTION TRAN (KSTAB,HEFF,NRFLAG,IBACK)          TRAN0000
C
C THIS FUNCTION CALCULATES THE COUPLING COEFFICIENT  TRAN0001
C AT A RECEPTOR DUE TO A POINT OR AREA SOURCE      TRAN0002
C
C COMMON /INFO/ IRECEP,IWNDIR,ITYPE      ,HTAERO,XS,YS,ZS,DELY,DELZ,  TRAN0003
C . TS,VS,DS,HS,PRFLAG,EMIS( 8 ),NPOL      TRAN0004
C COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,JSTAB,HMIX,TEMP,  TRAN0005
C . TDMY      TRAN0006
C COMMON /XTRAN/ CL,WSMD,TY,TZ      TRAN0007
C COMMON/WDUH/WSAVE      TRAN0008
C COMMON/LOC/DW,CW,HRECPT,WIT      TRAN0009
C DATA SQ2PI /2.5066283/      TRAN0010
C XL = CL      TRAN0011
C JSTABT=JSTAB      TRAN0012
C WSAVE=WSMD      TRAN0013
C WSKMPH=WSAVE*3.6      TRAN0014
C DELYKM=DELY/1000.      TRAN0015
C ADD1=0.      TRAN0016
C ADD2=0.      TRAN0017
C ADD3=0.      TRAN0018
C ADD4=0.      TRAN0019
C ADD5=0.      TRAN0020
C ADD6=0.      TRAN0021
C
C IF (KSTAB.GT.0) GO TO 121      TRAN0022
C
C KSTAB=0, PLUME IS BELOW THE LID, IF RECEPTOR IS ABOVE  TRAN0023
C LID, TRAN = 0.      TRAN0024
C
C IF (HRECPT.GT.HMIX) GO TO 76      TRAN0025
C HLID=HMIX      TRAN0026
C GOTO 140      TRAN0027
C
C ASSUME ARBITRARILY HIGH LID HEIGHT FOR TWO CASES WHEN:  TRAN0028
C KSTAB = 1, PLUME IS INITIALLY ABOVE THE LID      TRAN0029
C KSTAB = 2, PLUME WILL PENETRATE THE LID      TRAN0030
C ASSIGN STABILITY CLASSES 5 AND 4 RESPECTIVELY      TRAN0031
C
C 121 JSTABT=6-KSTAB      TRAN0032
C HLID=3050.      TRAN0033
C XL = 100.      TRAN0034
C
C 140 CONTINUE      TRAN0035
C DMIN=ABS(CW)      TRAN0036
C TT=TY+DW/WSKMPH      TRAN0037
C IF (TT.LE.0.) GO TO 76      TRAN0038
C IF (NRFLAG.NE.0) GO TO 143      TRAN0039
C
C NRFLAG=0, EFFECTS OF GROUND AND SKY LID ARE TREATED  TRAN0040
C BY THE MULTIPLE IMAGE METHOD, WITH UP TO 6 TERMS      TRAN0041
C INCLUDED IN THE COUPLING COEFFICIENT      TRAN0042
C
C TMIN = 0.      TRAN0043
C IF (DW-WIT.GT.0.) TMIN=(DW-WIT)/WSKMPH      TRAN0044
C IF (IBACK.EQ.0) GO TO 131      TRAN0045
C
C IBACK=1, RECEPTOR IS WITHIN CRITICAL DOWNWIND DISTANCE.  TRAN0046
C TREAT PORTIONS OF SOURCE UPWIND OF CRITICAL LENGTH      TRAN0047
C FROM RECEPTOR      TRAN0048
C
C IF (DW+WIT-DELYKM) 132,132,133      TRAN0049
C 132 TMIN=0.      TRAN0050
C
C

```

```

GO TO 131
133 TMIN=(DW+WIT-DELYKH)/WSKMPH           TRAN0062
131 TT=AMAX1(TMIN+TY,.01)                   TRAN0063
      SY=SIGY(JSTABT,TT)                     TRAN0064
      XRP = WIT + DW                         TRAN0065
      IF (XRP.LT.2.*WIT) DW=XRP/2.           TRAN0066
      TTT=TZ+DW/WSKMPH                      TRAN0067
      SZ=SIGZ(JSTABT,TTT)                    TRAN0068
      ED=DMIN/(SY/1000.)                    TRAN0069
      EXPD=-.5*ED*ED                      TRAN0070
      IF (EXPD.LT.-20.) GO TO 76            TRAN0071
      EDD=EXP(EXPD)                         TRAN0072
      IF (DW.GT.2.*XL) GO TO 153           TRAN0073
C
C   CALCULATE MULTIPLE IMAGES             TRAN0074
C
C   DENOM=6.2831853*WSAVE*SY*SZ          TRAN0075
C
C   IMAGES 1 AND 2: GROUND REFLECTION OF SOURCE TRAN0076
C
C   Z2=HEFF+HRECPT                      TRAN0077
C   Z1=HEFF-HRECPT                      TRAN0078
C   EXPZ=-(Z1/SZ)**2/2.                  TRAN0079
      IF (EXPZ.LT.-20.) GO TO 76          TRAN0080
      E1=EXP(EXPZ)                        TRAN0081
      ADD1=E1*EDD/DENOM                  TRAN0082
      IF (HEFF.GT.0.0) GO TO 171         TRAN0083
      ADD2=ADD1                          TRAN0084
      GOTO 172                           TRAN0085
171  CONTINUE                           TRAN0086
      EXPZ=-(Z2/SZ)**2/2.                TRAN0087
      IF (EXPZ.LT.-20.) GO TO 61          TRAN0088
      E2=EXP(EXPZ)                        TRAN0089
      ADD2=ADD1*E2/E1                     TRAN0090
C
C   IMAGES 3 AND 4: REFLECTION ABOUT HLID OF SOURCE TRAN0091
C
C   172  CONTINUE                         TRAN0092
      IF (HEFF.LE.HLID/2.0.AND.SZ.LT.(HLID-HEFF)/2.2) GO TO 61 TRAN0093
      IF (HEFF.LT.HLID) GO TO 174         TRAN0094
      ADD3=ADD1                          TRAN0095
      ADD4=ADD2                          TRAN0096
      GOTO 173                           TRAN0097
      CONTINUE                           TRAN0098
      Z=2.*HLID-Z2                      TRAN0099
      EXPZ=-(Z/SZ)**2/2.                  TRAN0100
      IF (EXPZ.LT.-20.) GO TO 61          TRAN0101
      E3=EXP(EXPZ)                        TRAN0102
      ADD3=ADD1*E3/E1                     TRAN0103
      Z=2.*HLID-Z1                      TRAN0104
      EXPZ=-(Z/SZ)**2/2.                  TRAN0105
      IF (EXPZ.LT.-20.) GO TO 61          TRAN0106
      E4=EXP(EXPZ)                        TRAN0107
      ADD4=ADD1*E4/E1                     TRAN0108
      IF (HEFF.GT.0.0) GO TO 173         TRAN0109
      ADD5=ADD3                          TRAN0110
      ADD6=ADD4                          TRAN0111
      GO TO 61                           TRAN0112
C
C   IMAGES 5 AND 6: REFLECTION ABOUT HLID OF FIRST BELOW GROUND IMAGE TRAN0113
C
C   173  CONTINUE                         TRAN0114
      ADD5=ADD4                          TRAN0115
      ADD6=ADD4                          TRAN0116
      GO TO 61                           TRAN0117
      ADD6=ADD4                          TRAN0118
      GO TO 61                           TRAN0119
      ADD6=ADD4                          TRAN0120
      GO TO 61                           TRAN0121
      ADD6=ADD4                          TRAN0122
      GO TO 61                           TRAN0123

```

```

Z=2.*HLID+Z1
EXPZ=-(Z/SZ)**2/2.
IF (EXPZ.LT.-20.) GO TO 61
E5=EXP(EXPZ)
ADD5=ADD1*E5/E1
Z=2.*HLID+Z2
EXPZ=-(Z/SZ)**2/2.
IF (EXPZ.LT.-20.) GO TO 61
E6=EXP(EXPZ)
ADD6=ADD1*E6/E1
GO TO 61
C
C UNIFORM MIXING ASSUMED
C
153 DENOM=SQ2PI*WSAVE*SY
ADD1=EDD/(DENOM*HLID)
61 CONTINUE
TRAN=ADD1+ADD2+ADD3+ADD4+ADD5+ADD6
RETURN
C
76 TRAN=0.
IBACK=0
RETURN
C
NRFLAG=1, RECEPTOR IS CLOSE TO SOURCE
C
143 IF (IBACK.EQ.1) GO TO 144
C
C UPWIND PART OF SOURCE
C
T2=(DW+WIT)/WSKMPH+TZ
DWDEDG=DW-WIT
GO TO 145
C
C DOWNWIND PART OF SOURCE
C
144 T2=TZ+XL/WSKMPH
DWDEDG=DW+WIT-DELYKM
145 IF (DWDEDG.LE.0.) GO TO 146
C
C RECEPTOR IS DOWNWIND . FIND TRANSIT TIMES FROM
C DOWNWIND EDGE OF SOURCE
C
T1=TZ+DWDEDG/WSKMPH
TT=TY+DWDEDG/WSKMPH
Z1=SIGZ(JSTABT,T1)
SY=SIGY(JSTABT,TT)
GO TO 147
C
C RECEPTOR IS UPWIND. FIND SHORTEST TRANSIT TIME FROM
C ANY PORTION OF SOURCE
C
146 T1=TZ
TT=TY
SZ1=DELYZ/2.4
SY=DELY/2.4
C
C COMPUTE COUPLING COEFFICIENT USING INTEGRATED
C EXPRESSION FOR 'NEAR' SOURCE
C
147 SZ2=SIGZ(JSTABT,T2)
155 ARGY=-(CW*1000./SY)**2/2.

```

```

IF (ARGY.LT.-20.) GO TO 76
B=ALOG (SZ1/SZ2)/ALOG (T1/T2)
EXZ1=EXP (-(HEFF-HRECPT)/SZ2)**2/2.)
EXZ2=EXP (-(HEFF+HRECPT)/SZ2)**2/2.)
EXY=EXP (ARGY)
IF (ABS(1.-B).LE..001) GO TO 2
FCNX=(T2/SZ2-T1/SZ1)*3600./(1.-B)
GO TO 3
C
C SPECIAL CASE FOR B = 1.
C
2 FCNX=T1/SZ1*ALOG (T2/T1)*3600.
C
3 TRAN=EXY*(EXZ1+EXZ2)*FCNX/(6.2831853*SY*DELY)
IBACK=0
RETURN
END

```

TRAN0186  
TRAN0187  
TRAN0188  
TRAN0189  
TRAN0190  
TRAN0191  
TRAN0192  
TRAN0193  
TRAN0194  
TRAN0195  
TRAN0196  
TRAN0197  
TRAN0198  
TRAN0199  
TRAN0200  
TRAN0201  
TRAN0202

#### REFERENCES

1. Bingaman, D. J. and L. E. Wangen, "Air Quality Assessment Model for Air Force Operations - Source Emissions Inventory Computer Code Documentation." Air Force Civil and Environmental Engineering Development Office report number, CEEDO-TR-76-33.
2. Menicucci, D. F., "Air Quality Assessment Model (AQAM) Data Reduction and Operations Guide," Air Force Weapons Laboratory report number AFWL-75-307, Oct 76.
3. Rote, Donald M., and L. E. Wangen, "A Generalized Air Quality Assessment Model for Air Force Operations - Technical Report," Air Force Weapons Laboratory report number AFWL-TR-74-304, February 1975.

INITIAL DISTRIBUTION

USAF/PREVP	3
TAC/DEEV	2
CINCUSAFA/Surgeon	1
USAFA/DEV	1
AFIT/DEM	1
OEHL/CC	1
OEHL/OL-AA	5
ADTC/CSV	1
DDC/TCA	12
USA Environ Hygn Agency	1
USA CERL	1
WESTDIV (Code 09BE)	1
Technology Transfer/EPA	1
R&D/EPA	1
NARF/Code 64270	2
Nav Post Grad Sch	2
EESD/DR	5
US Env Prt Agency	2
FAA/AEQ	1
Det 1 HQ ADTC/WE	2
Det 1 HQ ADTC/EC	1
Det 1 HQ ADTC/ECA	7
AFCEC/DEE	3
Dr Delaney	1
AUL/AUL-LSE-70-239	1
Det 1 HQ ADTC/PRL	2
AFATL/DLODL	1
AFATL/DLODR	1